Ancient Kannada Text Recognition using CNN: A Review

Anusha L S
Dept. of Electronics and Communication Engineering
R V College of Engineering
Bengaluru, India

Deeksha Sanu
Dept. of Electronics and Communication Engineering
R V College of Engineering
Bengaluru, India

Mounika Sri Ramesh Babu
Dept. of Electronics and Communication Engineering
R V College of Engineering
Bengaluru, India

Palak Chaudaha
Dept. of Electronics and Communication Engineering
R V College of Engineering
Bengaluru, India

***Abstract*— Kannada is a dravidian language most widely spoken in the indian state of Karnataka. With its rich collection of epigraphs like manuscripts and inscriptions, it is considered a repository of knowledge. Ancient text recognition using neural networks has emerged as a powerful and promising approach for deciphering and understanding historical manuscripts. Neural networks, particularly deep learning architectures like convolutional neural networks (CNNs) have shown remarkable capabilities in processing and analyzing visual and textual data. The application of neural networks in ancient text recognition has facilitated faster and more reliable transcription and translation processes, reducing the dependence on manual expertise and human error. It enables scholars and historians to efficiently explore vast collections of ancient texts, unveiling hidden knowledge and shedding light on historical civilizations. Although challenges such as script complexity, data scarcity, and document degradation persist, the integration of neural networks holds great promise in pushing the boundaries of ancient text recognition. By combining the power of artificial intelligence with historical research, neural network-based approaches are revolutionizing our understanding of ancient texts and enriching our knowledge of the past.**

Keywords—Ancient text; CNN; Neural networks; OCR; Character recognition

I. INTRODUCTION

Kannada is a Dravidian language spoken by over 60 million people in India. It is the official language of the state of Karnataka. The Kannada script is asyllabic, which means that each character represents a consonant or vowel sound. The script is written from left to right. As one of the oldest Dravidian languages, Kannada has a rich history of historic texts. Ancient Kannada text recognition using Convolutional Neural Networks (CNN) has emerged as a transformative approach in the field of historical studies. But traditionally, deciphering and interpreting these texts has required manual skill, which can be time-consuming and error-prone. The use of CNNs in the recognition of old Kannada texts has produced promising results recently, revolutionizing how these texts are understood and analyzed. Kannada poses a significant challenge for automated recognition systems because of its distinctive script and intricate characters. To accurately segment and recognise individual characters and words, advanced computational techniques are needed due to the script's complex structure, including its curves, loops, and stacked characters. The language has also evolved multiple times over the years, as shown in Fig 1., with the modern Kannada language being different from the ancient texts from the older dynasties. In order to recognise Kannada characters and text, CNNs, a subset of deep learning models, have demonstrated to be very successful in image processing tasks.

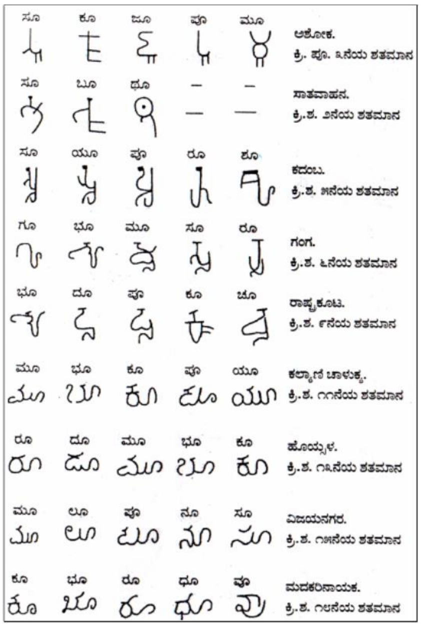

Fig. 1: The Evolution of Historical Kannada Handwritten Scripts of Different Dynasties [1].

The primary objective of ancient Kannada text recognition using CNNs is to develop automated systems that can accurately transcribe and translate ancient Kannada manuscripts. Large datasets of digitized Kannada texts representing a variety of scripts and linguistic variations are used to train these systems. These models learn to recognise and categorize the complex patterns and structures present in the old Kannada texts by utilizing the hierarchical feature extraction capabilities of CNNs. Researchers and academics can efficiently analyze and explore sizable collections of old Kannada texts by automating the transcription process, providing fresh insights into the language, literature, and cultural aspects of ancient Karnataka. Additionally, it democratizes access to this priceless knowledge, making a larger audience aware of it.

Although CNNs show great promise in the recognition of ancient Kannada text, there are still issues that need to be resolved. These include the need for specialized datasets for training and evaluation, script variation, the deterioration of old manuscripts, the lack of annotated training data, and the limitation of training data. To overcome these obstacles, it is necessary for computer scientists, linguists, and historians to work together to create reliable recognition models that are specifically designed for the peculiarities of old Kannada texts.

II. History of Kannada Text Recognition

Ancient Kannada inscriptions and manuscripts date back to as early as the 5th century CE. These inscriptions were typically carved on stone or metal, while manuscripts were written on palm leaves or copper plates. Deciphering and recognizing the text in these ancient materials required expertise in epigraphy, paleography, and linguistic analysis. The first attempts at Kannada text recognition were made in the early 1980s. These early systems were based on rule-based approaches. In the late 1990s, neural network-based approaches were developed. These systems were more accurate than rule-based systems, but they were also more complex and computationally expensive. The development of optical character recognition (OCR) technology, which is the process of transforming scanned or photographed images of text into editable and searchable digital text, can be credited with the development of Kannada text recognition. Due to the complexity of the Kannada script, developing OCR technology for it posed special difficulties. Each character in the alphasyllabary of the Kannada script denotes a syllable. There are many different characters in it, including vowels, consonants, and many different combinations. Conjunct consonants and ligatures also make the process of recognition more challenging.

Due to a lack of information, resources, and computing power, early attempts at Kannada text recognition were unsuccessful. However, progress has been made in Kannada OCR over time thanks to technological developments and the availability of larger datasets. Organizations and researchers have concentrated on creating algorithms and models that are uniquely suited to the Kannada script [2]. To increase the accuracy of Kannada text recognition, researchers have looked into a number of techniques, including pattern recognition, machine learning, and neural networks. Additionally, the advancement of open-source tools and resources has aided Kannada OCR. These tools give researchers and programmers a platform for cooperation, knowledge exchange, and enhancement of Kannada text recognition systems. The field of Kannada text recognition is anticipated to continue developing due to ongoing research and technological advancements, making it simpler to digitise and process Kannada texts for a variety of purposes, including automated data extraction, content analysis, and the digitization of historical documents.

III. Challenges in Ancient Kannada Text Recognition

Recognizing and deciphering ancient Kannada texts presents several challenges due to various factors, including the script's complexity, the preservation state of the texts, and the lack of available resources. Here are some of the key challenges in ancient Kannada text recognition:

A. *Script Evolution*

The Kannada script has evolved over time, with changes in letter forms, orthographic conventions, and ligatures. Recognizing and understanding the script's different stages and variations is essential for accurately deciphering ancient texts.

B. *Lack of Standardization*

Ancient Kannada texts often lack standardized spelling and grammar. Variations in the usage of characters, ligatures, and abbreviations can pose challenges for recognition. The same word or phrase may be written in different ways, making it difficult to establish consistent patterns.One of the most striking examples of the lack of standardization in ancient Kannada texts is the variation in the spelling of vowels. In some texts, the vowels are written with diacritics, while in others they are not. This can make it difficult to determine the correct pronunciation of words.

C. *Damaged and Faded Texts*

Many ancient Kannada inscriptions and manuscripts have deteriorated over time, resulting in damaged or faded text. The loss of characters or incomplete preservation can make it challenging to reconstruct the original content and accurately recognize the text. Recognizing and deciphering damaged and faded ancient Kannada texts poses additional challenges in the field of text recognition. However, advancements in imaging technology, computational methods, and historical research have provided some approaches to address these challenges [3].

D. *Regional Variations*

Regional variations in ancient Kannada text recognition refer to the differences in the script, writing styles, and orthographic conventions found across different regions and time periods within the Kannada-speaking region. These regional variations pose challenges in recognizing and interpreting ancient Kannada texts accurately. Ancient Kannada texts span various historical periods, each with its own cultural, social, and political context. Regional variations in script and writing styles can be influenced by specific historical events or cultural practices. Understanding the historical context of a text can aid in deciphering regional variations and providing accurate interpretations.

E. *Limited Available Resources*

The availability of references, dictionaries, and corpora specifically dedicated to ancient Kannada texts can be limited. Ancient Kannada texts may be fragmented or incomplete due to various reasons such as erosion, damage over time, or improper preservation. The absence of comprehensive databases or repositories that systematically collect and catalog ancient Kannada texts can make it difficult for researchers to access and analyze a wide range of texts. A lack of centralized resources makes it challenging to conduct extensive research and comparative analysis.

F. *Epigraphic Conventions*

Epigraphic conventions, such as layout, abbreviations, and contextual information, can vary across regions. These conventions may differ in different time periods and inscriptions from different locations. Familiarity with regional epigraphic conventions is necessary for accurate recognition and interpretation of ancient Kannada texts.

G. *Specialized Knowledge and Expertise*

Recognizing ancient Kannada texts requires expertise in epigraphy, paleography, linguistics, and historical context. It often involves interdisciplinary collaboration between experts in these fields to overcome the challenges posed by the complex nature of the texts.

Despite these challenges, scholars, epigraphists, and researchers have made significant progress in recognizing and deciphering ancient Kannada texts. The collaboration between experts, advancements in imaging technology, and the accumulation of linguistic knowledge have contributed to the gradual decipherment and understanding of these invaluable historical and cultural artifacts.

IV. CNN FOR ANCIENT KANNADA TEXT RECOGNITION

Convolutional Neural Networks (CNNs) have been successfully employed in various applications of computer vision, including text recognition. While CNNs have primarily been used for modern text recognition, they can also be adapted for ancient Kannada text recognition with certain considerations. An example of a typical convolution neural network is shown in Fig. 2.

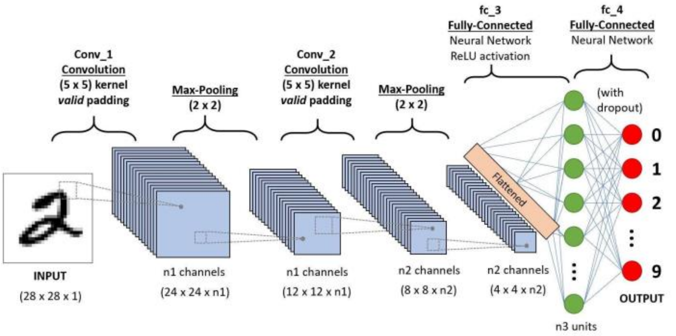

Fig. 2: A sample CNN network [4].

There are various CNN architectures that can be effective for ancient Kannada text recognition. LeNet-5, is one of the early CNN architectures designed for handwritten digit recognition. It consists of convolutional and pooling layers followed by fully connected layers. LeNet-5 can serve as a starting point for recognizing individual characters or digits in ancient Kannada text. VGGNet, with its uniform architecture and multiple convolutional layers, can be adapted for ancient Kannada text recognition. By training the network on Kannada text datasets, VGGNet can learn to extract features relevant to Kannada characters and achieve good recognition performance. ResNet, known for its residual connections, can be applied to tackle the challenges of recognizing degraded or complex ancient Kannada text. The deeper architecture of ResNet allows for better representation learning and can capture intricate details and variations in the text images. DenseNet: DenseNet's dense connectivity pattern, where each layer is connected to every other layer, promotes feature reuse and can be beneficial for recognizing ancient Kannada text. The dense connections enable the network to effectively capture the long-range dependencies and intricate structures within the text. Inception/GoogLeNet with its inception modules and parallel convolutional layers of different sizes, can be utilized to capture different scales and variations in ancient Kannada text. The inception modules help in capturing both local and global features, which can be advantageous for recognizing characters in varied styles and degradation levels. Transformer architectures, such as the popular BERT (Bidirectional Encoder Representations from Transformers), have been successful in various natural language processing tasks. These models can be adapted for ancient Kannada text recognition by treating the text as a sequence of characters or subwords and leveraging the self-attention mechanism to capture contextual information.

It's important to note that the performance of CNN architectures for ancient Kannada text recognition will depend on the availability and quality of the training data, the complexity and degradation levels of the text, and other specific challenges associated with ancient scripts. Experimentation and fine-tuning of the chosen architectures may be required to achieve optimal results. Typically when using CNN, the following steps are followed:

A. *Dataset Collection*

Building a dataset of ancient Kannada text samples is a crucial step. This dataset should include a diverse range of ancient Kannada inscriptions and manuscripts, from various dynasties, capturing variations in script, writing styles, and preservation conditions. Collecting a substantial and representative dataset is essential for training a robust CNN model. It operates by employing convolutional layers to extract meaningful features from input data, such as images, and using pooling layers to reduce spatial dimensions. These extracted features are later passed through fully connected layers for final classification or regression.

B. *Preprocessing*

Preprocessing in CNN involves transforming and preparing the input data before feeding it into the network for training or inference. Preprocessing plays a vital role in preparing the ancient Kannada text images for CNN recognition. This may involve image cleaning, denoising, normalization, and enhancing contrast to improve the quality of the text images. Additionally, techniques like binarization can be used to convert the images into binary representations for further processing. Some of the commonly used preprocessing techniques are:

1) *Gray Scaling:* In order for the computer to display saturation, images are nothing more than neat arrays of 0 to 255 numeric values. Red, green, and blue are fundamentally mixed in different ratios to create color. An array of red, blue, and green values is digitally stored when an image is a color image. For color outputs, the computer "mixes" these on the spot. The neural network performs convolution on each of the red, green, and blue channels as soon as it notices this [5]. However, this makes our calculation more difficult and one needs to perform convolutions on three different arrays rather than just one array for a single convolution. By downscaling an image's 3D pixel value to a 1D value, the complexity of the image is decreased.
2) *Binarization:* In the image processing stage for document image scanning and processing, the thresholding technique plays a significant role. Binarization's main goal is to take the original images' characters and remove any noise [6]. In the context of degraded document images, researchers and practitioners have proposed various binarization techniques to improve the readability and analysis of such documents. These techniques typically aim to overcome challenges posed by degradation factors and enhance the contrast between foreground text and background. Some common binarization techniques for degraded document images include global thresholding, local thresholding, adaptive thresholding and hybrid thresholding [7]. The goal of these binarization techniques is to accurately separate the foreground text from the background in degraded document images, enabling better document analysis, text recognition, and information extraction.
3) *Cropping and Resizing:* Cropping and resizing are common image processing techniques used to modify the size and composition of images. Cropping refers to the process of selecting and extracting a specific region of interest (ROI) from an image while discarding the remaining parts. This operation helps focus on the most relevant content and remove unwanted or distracting elements. Resizing involves changing the dimensions (width and height) of an image while maintaining its aspect ratio. This operation is commonly used to standardize the size of images in a dataset or to match the input requirements of a machine learning model. Both cropping and resizing can be performed using various programming libraries and software tools, such as OpenCV, PIL (Python Imaging Library), or image editing software like Adobe Photoshop or GIMP. These operations are flexible and can be customized to suit the specific needs of an application, dataset, or machine learning model.
4) *Edge detection:* Edge detection algorithms identify and highlight the boundaries or edges of objects in an image. In the case of text recognition, edge detection can help in extracting the textual content from the background and enhancing the visibility of the text for further processing [8]. There are several edge detection algorithms available, and the choice depends on the specific requirements and characteristics of the Kannada text images. Some commonly used edge detection algorithms include:

   (i) Canny Edge Detection: The Canny edge detector is a popular choice due to its ability to accurately detect edges while suppressing noise. It involves multiple stages, including noise reduction, gradient computation, non-maximum suppression, and hysteresis thresholding [9].

   (ii) Sobel Operator: The Sobel operator is a simple and efficient algorithm that calculates gradients in the horizontal and vertical directions. It highlights regions of significant intensity changes, which often correspond to edges [10].
5) *Smoothing and Sharpening:* Smoothing, also known as blurring, is a technique used to reduce the presence of noise or fine details in an image, resulting in a smoother appearance. It helps to eliminate high-frequency components that may be caused by sensor noise, compression artifacts, or other factors. Smoothing can also be useful for reducing image noise or removing unwanted textures [11]. Sharpening is a technique used to enhance the edges and details in an image, increasing its apparent sharpness and clarity. It works by emphasizing high-frequency components to enhance the transitions between different image regions. Sharpening can be particularly useful when dealing with blurry or low-quality images.

C. *Line Segmentation and Character Segmentation*

In ancient Kannada texts, characters are often densely connected or ligatured, making character segmentation challenging. Specialized techniques need to be employed to segment characters accurately from the text images. This may involve analyzing contours, connected components, or applying graph-based algorithms to separate characters. Robust and effective page segmentation is required in order to achieve accurate text recognition performance for historical handwritten document images [12]. At this level of segmentation, the image with text that is written in lines is presented to us. To segment the image into lines is the goal of line level segmentation. The concept is that the rows that represent the text in a line have a higher number of foreground pixels, which in turn corresponds to higher peaks in the histogram when the binary image is being horizontally projected. High numbers of background pixels are present in the rows that represent the spaces between the lines, which correspond to lower histogram peaks. To divide the lines, it is possible to choose rows as the segmenting lines that correspond to the histogram's lower peaks. The process of character segmentation breaks down an image of a string of characters into smaller images representing individual symbols. It is a decision-making process in an optical character recognition (OCR) system. Its conclusion that an isolated pattern in the image represents a character (or another identifiable unit) may be accurate or inaccurate. It makes a significant contribution to the system's error rate by being incorrect frequently enough. At this stage of segmentation, we are given an image containing a single word that was previously segmented and is composed of a string of characters. Thus, text line segmentation is crucial to a document recognition system's overall effectiveness.

The optical character recognition (OCR) system's character segmentation is a crucial step. One of the important elements that affects how well a recognition system performs is the segmentation of broken characters. Character level segmentation aims to divide the text in the image into distinct characters. Depending on the situation in which the OCR is being used, this level of segmentation is optional. Character Level Segmentation is not necessary if the OCR technology is being used on text when words have independent characters. We can segment the characters in the previous stage (by choosing a very low threshold) since a uniform gap is kept between the characters within a word, whatever how small it may be. On the other hand, character level segmentation must be carried out if the OCR system is being applied to the text and characters within a word are joined (Cursive handwriting). The segmentation procedure uses geometry and shape to locate the segmentation points and word image thinning to retrieve the width of a pixel's worth of stroke and find the ligatures of Kannada characters. With respect to Kannada scripts, character level segmentation is extremely important in identifying the characters which are more closely placed and complex [13]. The characters which are now segmented can be identified, in modern day kannada and then be processed, classified and stored accordingly.

D. *Training CNN Models*

CNN models can be trained using the preprocessed dataset. The architecture of the CNN can be designed to accommodate the specific requirements of ancient Kannada script recognition. It may include multiple convolutional layers, pooling layers, and fully connected layers[14], [15]. However, there are various types of CNN (Convolutional Neural Network) architectures, each with its own characteristics and applications. Typically training CNN models involves the following steps:

1) *Split the dataset into training, validation, and test sets:* The training set should be the largest portion, typically around 60-80% of the dataset, to provide sufficient data for learning. Allocate a smaller portion, usually around 10-20%, for both the validation and test sets.
2) *Initialize the chosen CNN model with random or pretrained weights on a related task:* For ancient Kannada text recognition, the availability of pretrained weights specifically trained on Kannada text may be limited. Therefore, initializing the chosen CNN model with random weights is a common approach. Initialize the weights of the CNN layers randomly. This can be done by drawing the weights from a normal distribution or a uniform distribution centered around zero. By initializing the chosen CNN model with random weights, the model starts with no prior knowledge of Kannada text features and learns to extract relevant features from the training data.
3) *Train the model using the training set:* The model is trained by feeding batches of images through the network, computing the loss, and adjusting the model's weights through backpropagation and optimization techniques (e.g., stochastic gradient descent).
4) *Monitor the model's performance:* The model’s performance and accuracy is monitored on the validation set and adjustments are made as needed, such as modifying hyperparameters or using techniques like learning rate scheduling or early stopping to prevent overfitting.

E. *Data Augmentation*

To enhance the robustness of the CNN model, data augmentation techniques can be employed. This involves applying transformations such as rotation, scaling, translation, or adding noise to the training images. Data augmentation helps the model generalize better and handle variations in the ancient Kannada script. It's important to apply these data augmentation techniques with caution and consider the specific characteristics of ancient Kannada text. Ensure that the augmented images still maintain the integrity and legibility of the text [16]. Experimentation and iterative testing with different augmentation techniques can help determine the most effective combination for improving the model's performance on the ancient Kannada text recognition task. Some of the common data augmentation techniques for ancient Kannada text recognition are:

1) *Rotation:* Rotate the images by a certain angle (e.g., -15 to +15 degrees) to simulate variations in the orientation of the text. This helps the model learn to recognize characters from different angles.
2) *Translation:* Shift the images horizontally or vertically by a certain amount to simulate variations in the position of the text within the image. This helps the model become invariant to small shifts in the text's location.
3) *Scaling:* Resize the images by scaling them up or down within a certain range. This can simulate variations in the font size or different text sizes in the ancient Kannada documents.
4) *Shearing:* Apply a shearing transformation to the images, which tilts the text at different angles. This can mimic the distortion or slant often found in ancient text.
5) *Flipping:* Flip the images horizontally or vertically to introduce mirror reflections. This helps the model learn invariant representations of characters, regardless of their orientation.
6) *Noise injection:* Add random noise (e.g., Gaussian noise) to the images to simulate variations in image quality or degradation commonly seen in ancient documents.
7) *Contrast and brightness adjustment:* Adjust the contrast and brightness of the images to simulate differences in lighting conditions or to normalize the overall image intensity.
8) *Elastic deformation:* Apply elastic deformations to the images by locally distorting small image patches. This can simulate the warping or distortions often found in ancient text images.

F. *Model Evaluation and Optimization*

The trained CNN model needs to be evaluated using appropriate metrics and validation techniques. Performance measures such as accuracy, precision, recall, and F1-score can be used to assess the model's recognition capabilities. Fine-tuning and hyperparameter optimization techniques can be employed to improve model performance. The evaluation metrics to assess the strengths and weaknesses of the model need to be analyzed. Any potential issues, such as underperformance on specific classes or misclassifications of certain text patterns should be identified. A detailed error analysis to gain insights into the model's mistakes and patterns of misclassifications can be conducted.The model can then be fine tuned based on the evaluation and error analysis results. Common optimization techniques include:

1) *Adjusting hyperparameters:* Experiment with different learning rates, batch sizes, and regularization techniques (e.g., dropout, L1/L2 regularization) to find optimal settings.
2) *Architecture modifications:* Explore changes to the model architecture, such as adding or removing layers, increasing model depth, or adjusting filter sizes, to enhance its capability to recognize ancient Kannada text.
3) *Ensembling:* Combine multiple models or use ensemble techniques (e.g., averaging predictions) to leverage the collective knowledge of different models and improve overall performance.

G. *Post-processing*

Post-processing steps are performed to refine the output of the CNN model. This may involve language-specific rules and heuristics to correct errors, handle ligatures, and reconstruct the original text. Linguistic knowledge and historical context can guide post-processing techniques to improve the accuracy of the recognized text [17]. For example, Kannada characters often undergo ligature formation, where multiple characters merge into a single ligature. Post-processing can include handling such ligatures or splitting them into individual characters based on contextual information. Language models, such as n-gram models or recurrent neural networks (RNNs), can be used to predict the likelihood of word sequences and improve the accuracy of the recognized text based on language patterns and contextual information.

It's important to note that due to the complex nature of ancient Kannada text and the challenges specific to script variations and preservation conditions, the accuracy of CNN-based recognition systems may vary. The adaptation of CNNs for ancient Kannada text recognition requires a combination of expertise in computer vision, linguistics, epigraphy, and domain-specific knowledge to overcome the challenges posed by the ancient script.

V. RECENT ADVANCES TO IMPROVE ACCURACY

Recent advances in CNNs have led to improved accuracy and robustness in ancient Kannada text recognition. This has made it possible to automatically recognize ancient Kannada text with high accuracy. This is a significant advance, as it allows researchers to access and study a vast body of ancient Kannada literature that was previously inaccessible. Some of the recent advances in CNN-based text recognition that could potentially be applied to ancient Kannada text recognition are:

A.*Advanced CNN Architectures*

CNN architectures have evolved with advancements such as deeper networks and attention mechanisms (e.g., Transformer) that capture complex features and dependencies within the text. These architectures can potentially improve the recognition accuracy and robustness for ancient Kannada text by capturing fine-grained details and long-range dependencies in the script.

B.*Transfer Learning and Pretraining*

Transfer learning techniques, such as using pretrained models on large-scale text recognition tasks, can benefit ancient Kannada text recognition if pretrained models on similar scripts or languages are available. By leveraging the learned representations from large datasets, the model can generalize better and achieve improved performance even with limited labeled ancient Kannada text data.

C.*Attention Mechanisms*

Attention mechanisms can be incorporated into CNN models to selectively focus on relevant parts of the ancient Kannada text, enabling the model to attend to important details and improve recognition accuracy. Attention-based models can be particularly useful when dealing with long sequences or complex scripts like ancient Kannada.

D.Sequence Modeling

Instead of recognizing text at the character or word level independently, sequence modeling approaches like Connectionist Temporal Classification (CTC) or Transformer-based models can be explored to capture dependencies between adjacent characters or words in the ancient Kannada text.

VI. CONCLUSION

In conclusion, CNN models have shown promise in the field of ancient Kannada text recognition. While there may not be specific research on recent advances in this area, the general advancements in CNN architectures, preprocessing techniques, data augmentation, attention mechanisms, transfer learning, and sequence modeling can be applied to improve the recognition accuracy and robustness of ancient Kannada text recognition systems. By leveraging CNN models, it becomes possible to automatically recognize and interpret ancient Kannada text, which is often challenging due to variations in script styles, degradation, and limited availability of labeled data. CNN models excel at learning hierarchical features and capturing spatial dependencies, making them suitable for extracting meaningful representations from ancient Kannada text images.

REFERENCES

[1] Bannigidad, Parashuram & Gudada, Chandrashekar. (2019). Historical Kannada Handwritten Character Recognition using K-Nearest Neighbour Technique.

[2] Indira, K. & Selvi, S. Sethu. (2010). Kannada Character Recognition System A Review.

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

[4] [1] S. Bhardwaj, “Convolutional Neural Networks : Understand the basics,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-understand-the-basics/ (accessed May 23, 2023).

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

[6] P. Ranjitha and T. D. Shreelakshmi, "A Hybrid Ostu based Niblack

Binarization for Degraded Image Documents," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-7, doi: 10.1109/INCET51464.2021.9456150.

[7] Panchal, Amoli & Panchal, Chintan & Shah, Bhargav. (2017). Image binarization techniques for degraded document images: A review Binarization techniques.

[8] S. Wang, K. Ma and G. Wu, "Edge Detection of Noisy Images Based on Improved Canny and Morphology," 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2021, pp. 247-251, doi: 10.1109/ECICE52819.2021.9645601.

[9] Y. Li and B. Liu, "Improved edge detection algorithm for canny operator," 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2022, pp. 1-5, doi: 10.1109/ITAIC54216.2022.9836608.

[10] Y. Zhang, X. Han, H. Zhang and L. Zhao, "Edge detection algorithm of image fusion based on improved Sobel operator," 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2017, pp. 457-461, doi: 10.1109/ITOEC.2017.8122336.

[11] Li, Pei & Wang, Hongjuan & Yu, Mengbei & Li, Yeli. (2021). Overview of Image Smoothing Algorithms. Journal of Physics: Conference Series. 1883. 012024. 10.1088/1742-6596/1883/1/012024.

[12] B. Gatos, G. Louloudis and N. Stamatopoulos, "Segmentation of Historical Handwritten Documents into Text Zones and Text Lines," 2014 14th International Conference on Frontiers in Handwriting Recognition, Hersonissos, Greece, 2014, pp. 464-469, doi: 10.1109/ICFHR.2014.84.

[13] C. Naveena and V. N. Manjunath Aradhya, "Handwritten character segmentation for Kannada scripts," 2012 World Congress on Information and Communication Technologies, Trivandrum, India, 2012, pp. 144-149, doi: 10.1109/WICT.2012.6409065.

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

[15] K. Asha and H. K. Krishnappa, "Kannada Handwritten Document Recognition using Convolutional Neural Network," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 299-301, doi: 10.1109/CSITSS.2018.8768745.

[16] A. Spruck, M. Hawesch, A. Maier, C. Riess, J. Seiler and A. Kau, "3D Rendering Framework for Data Augmentation in Optical Character Recognition : (Invited Paper)," 2021 International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 2021, pp. 1-4, doi: 10.1109/ISSCS52333.2021.9497438.

[17] A. İ. Topçu and B. Uğur Töreyin, "Neural Machine Translation Approaches for Post-OCR Text Processing," 2022 30th Signal Processing and Communications Applications Conference (SIU), Safranbolu, Turkey, 2022, pp. 1-4, doi: 10.1109/SIU55565.2022.9864878.