IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

Handwritten Character Recognition of Kannada Language Using Convolutional Neural Networks

Sunil Kumar B L[1], Ananya Bhat[2], K Vinayaka Bhat[3], Chaitra S[4], Kruthi A Suvarna[5]

Professor, dept of Computer Science and Engineering, Canara Engineering College Mangalore,

Karnataka, India, 574219[1]

Student, dept of Computer Science and Engineering, Canara Engineering College Mangalore,

Karnataka, India, 574219[2-5]

Abstract: Handwritten Character Recognition of Kannada Language using Convolutional Neural Network is the project aimed at preserving the handwritten script in digital format, particularly for Kannada language, since Kannada is a language that is spoken by almost all of the residents of Karnataka. Kannada language has 49 base characters which include 15 vowels and 34 consonants. The project focuses on converting handwritten characters or sentences by recognizing them and converting them into digital format using Convolutional Neural Network (CNN) technology. To determine the model's effectiveness, it must first be trained, then validated and ultimately tested. The model gave a prediction performance of 97 % for the testing set. This result highlights the potential of this particular project to significant improvements in the accessibility to historical records and the streamlining of administrative processes through the successful implantation of the handwritten character recognition system.

Keywords: Convolutional Neural Network, Handwritten Character Recognition, Artificial Intelligence, Optical Character Recognition.

I. INTRODUCTION

Keyboards are the standard interface used to communicate with computers and send commands. Since the character set of Indian languages is somewhat complicated, keyboard input is essentially more appropriate for English than for Indian languages. This is because, in most situations many combinations of symbols are required to express a single character. This leads to the demand for giving input using natural handwriting. Handwritten Kannada Recognition is the method which recognizes the Kannada character which is submitted in the form of an image as the input. The Kannada language holds a very special place in the culture of the state and linguistic diversity of our country India. Kannada is Dravidian language predominantly spoken by residents of Karnataka [1]. The Kannada language evolved from the 5th century Kadamba script, which is usually written using the Kannada script [2]. Kannada has 49 base characters which include 15 vowels and 34 consonants. Handwritten character recognition using CNN technology outperforms alternative recognition methods, which frequently depends on recognition of characters and converting them into digital format [3]. The handwritten text recognition system is classified into the offline system and online system [4].

The main goal of the proposed study is to create an offline handwritten recognition system that can appropriately identity handwritten Kannada scripts and convert them into digital format. Sometimes a person writes a character differently than before, so collecting diverse samples with various writing styles, sizes and orientation is important. It is a known fact that the accuracy of the model could be highly affected by various factors like the quality of the image and handwriting style. Various preprocessing steps like resizing the grayscale conversion methods will be applied to the input image to improve the model performance. CNN will be utilized to extract features from the pre-processed image.

The desired features from the image will be extracted using various strategies during feature extraction, including convolutional operation, max pooling, flattening and fully connected layer. Feature detectors are used in the convolutional operation to learn and map the image's features, maintaining the relationship between the pixels. Depending on the size of the image, the Max Pooling layer will reduce the number of parameters. Flattening technique would help in relevant feature extraction by converting it into a one-dimensional vector, which is later classified. Using Handwritten Character Recognition as key technology to automate the recognition of characters from input images, increases the digitization process's efficiency. The project's dedication to pushing the technology concerning the handwritten character recognition (HCR) program highlights the innovative methods and developments it offers to improve character recognition technology.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

II. LITERATURE SURVEY

[1] Neural network and TensorFlow techniques are used for Handwritten Character Recognition (HCR). It has a thorough process for character recognition. Character recognition and classification are two applications of the SoftMax Regression approach. Results are improved by using direction and diagonal feature extraction techniques. Utilized is a feed forward model that was trained using a back-propagation approach. Normalization increased recognition accuracy.
This is probably going to aim high accuracy in recognizing handwritten characters using this specific approach. It might also explore using SoftMax Regression to assign probabilities to characters being recognized. This typically involves an input layer representing the image pixels, hidden layers that process the data, and an output layer with units corresponding to the characters the network can recognize. The network is trained using backpropagation. This involves feeding the network character images with known labels, calculating the output, and then adjusting the network's weights to minimize the difference between the actual output and the desired output.

[2] Optical character Recognition using back propagation neural network here, the English characters are recognized and categorized using feed-forward neural networks. In this case, the datasets include alphanumeric characters, tiny, capital, and lowercase letters. When the model was applied independently to each set, as opposed to integrating all sets at once, better results were seen.

[3] An Efficient Approach for Handwritten Devanagari character Recognition based on Artificial Neural Network involves utilizing Histogram of Oriented Gradients (HOG), which is structured into segmentation, preprocessing, features extraction, and classification stages. Here, Artificial Neural Network (ANN) are utilized to classify the characters and feature extraction is executed by partitioning the image. Devanagari script is the primary writing system used for Hindi, Marathi and Sanskrit languages. It presents unique challenges for character recognition due to its complex character shapes and ligatures. Techniques the handwritten Devanagari character images for the ANN includes noise reduction, normalization and segmentation.

[4] Improved Optical Character Recognition with Deep Neural Network presents a novel approach for enhancing optical character recognition (OCR) using deep neural networks (DNNs). OCR systems are essential for converting scanned documents or images containing text into editable and searchable digital text. In this paper, the authors propose leveraging the capabilities of deep neural networks, specifically convolutional neural networks (CNNs), to improve the accuracy and efficiency of OCR systems. The approach involves training a DNN model on a large dataset of labeled images containing various fonts, styles, and sizes of characters. Through the process of supervised learning, the DNN learns to extract hierarchical features from the input images, enabling it to effectively recognize characters even in the presence of noise or variations in writing styles.

[5] Rotation-Free Online Handwritten Chinese Recognition using Two- Stage Convolutional Neural Network presents an innovative approach to online handwritten Chinese character recognition that is robust against character rotation. Unlike traditional offline recognition systems that process static images, online recognition systems analyze dynamic strokes as they are drawn, making them suitable for real-time applications such as digital handwriting input. In this paper, the authors propose a two-stage convolutional neural network (CNN) architecture tailored specifically for online handwritten Chinese characters. The first stage involves stroke segmentation and feature extraction, where the CNN learns to identify and extract relevant stroke features from the input trajectory data. The second stage focuses on character classification, where the extracted features are fed into another CNN to recognize the corresponding Chinese characters.

[6] Kannada Handwritten Script Recognition using Machine Learning Techniques the recognition of Kannada characters employs two different techniques. In the first method, the Tesseract tool processes an image containing a character in various combinations. Here the image undergoes pre-processing, transitioning to grayscale and employing denoising techniques for noise removal. Meanwhile, the second technique, Convolutional Neural Network (CNN), necessitates breaking down words into individual characters before pre-processing. A letter count is established to determine x and y coordinates, and the characters are segmented and stored in respective folders. The Tesseract tool achieves an accuracy of 86%, while CNN achieves a slightly higher accuracy of 87%.

[7] Cursive Handwritten Text Recognition Using Bi-Directional LSTM: A Case Study on Urdu Handwriting Cursive handwritten text recognition is performed in the language Urdu. The dataset used is UNHD. There are two steps involved in the character recognition namely CNN for feature extraction and bi-directional Long-Short term memory technique for the classification. CNN uses seven layers and is followed by a B-LSTM layer. An accuracy of more than 83% overall is obtained. LSTM networks are well-suited for handling sequential data and capturing long-range dependencies, making them suitable for tasks like handwriting recognition.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

[8] Introducing the Boise State Bangla Handwriting Dataset and an Efficient Offline Recognizer of Isolated Bangla Characters, character recognition is performed for Bangla language. The dataset used is Boise State Bangla Handwriting Dataset. It involved segmentation of characters and x and y coordinates marking the length and height of the character is obtained. Features are classified as zonal, pattern and gradient features. Support Vector Machine (SVM) is used for classification purposes. Accuracy is improved by One Versus One (OVO) technique. Here the accuracy of 96.42% of accuracy is obtained. The creation of the Boise State Bangla Handwriting Dataset and the development of an efficient offline recognizer for isolated Bangla characters.

[9] Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition, Gurmukhi is an Indic language used in Punjab. It has 35 letters and 10 numerals. The dataset was collected from people who wrote the language. The grey scale conversion of the image is done before skeletonization where letters are given proper structure and normalized. The image undergoes segmentation to segregate the letters and the features that are extracted. An accuracy of 98.06% is obtained from this. The proposed technique focuses on feature extraction, a crucial step in character recognition systems, where relevant information is extracted from input images to enable classification. Horizontal projection captures the distribution of ink along the rows of the character image, while vertical projection captures the distribution along the columns.

[10] Handwritten Kannada numerals recognition using deep learning convolution neural network (DCNN) classifier, In the realm of pattern recognition, the task of identifying handwritten Kannada numerals is tackled in this paper. The focus is on establishing a robust process for recognizing Kannada numerals. Handwritten Kannada characters, presented in a document-like format, undergo pre-processing and attribute extraction. Pre-processing involves essential steps such as noise removal, normalization and thinning. Features are extracted using techniques like Drift Length Count, Discrete Wavelet Transform (DWT), and Curvelet Transformation Wrapping. To enhance the classification process, a deep convolutional neural network classifier is employed. The anticipated isolation accuracy for Kannada numerals in this approach is an impressive and that is 96%.

III. METHODOLOGY

Datasets

The dataset is collected from "Dhruvil Dave. (2021). Kannada Handwritten Characters [data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/1972687"or'https://www.kaggle.com/datasets/dhruvildave/kannada characters/code'

The dataset used contains 16,425 images of handwritten characters in the Kannada language to train and test the model. The training and test set distribution is 70% train images and 30% test images. It is a classification dataset that can be used for Computer Vision takes. It contains 657 classes with 25 images of each class. The dataset used to train the model is shown in Figure.1.

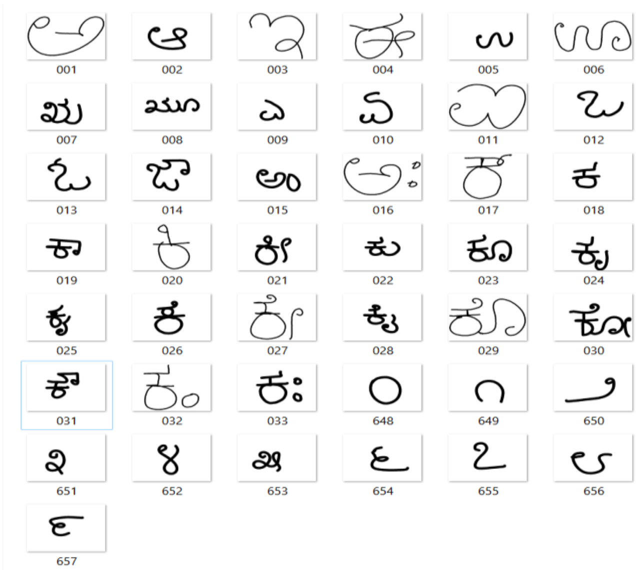

Figure 1 Dataset

IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

Preprocessing

The dataset is pre-processed to prepare it for input into Convolutional Neural Network (CNN) model. The images are read in grayscale mode, converting it to a single-channel image. After loading the image, the thresholding function applies a binary thresholding technique that converts the grayscale image into a binary image by setting pixel values above a certain threshold to 255(white) and those below the threshold to 0(black). The threshold value used here is 128. Then the binary image is resized to ensure that all images fed into CNN have consistent dimensions, which is essential for model training and inference. Finally, the scaled image's pixel values are normalized by dividing them by 255.0. Normalization scales the pixel values to a range between 0 and 1, which helps improve the convergence and stability of the neural network during training.

Overall, the input images are converted to a standardized format suitable for feeding into the CNN model, facilitating effective training and inference.

CNN Model

Convolutional Neural Networks (CNNs) are the class of deep neural networks commonly applied to analyzing visual imagery. CNN is used to extract the features after preprocessing the images. CNNs have revolutionized various fields, especially computer vision tasks, including image recognition, object detection, image segmentation, and more. A CNN model is created where layers are sequentially stacked from the input layer, all through the intermediate layers to the output layer.

1. Input Layer: The input layer acts as its entry point for data. The format of input data that the CNN model expects to receive is grayscale images with a size of 64x64 pixels.

2. Convolutional Layers: CNNs consist of multiple layers, with the first few layers usually being convolutional layers. The second convolutional layer has 32 filters of size 3x3. The second convolutional layers ReLU (Rectified Linear Unit) activation function is used for introducing non-linearity, allowing the network to learn more complex features.

3. Pooling Layers: Pooling layers reduces the spatial dimensions (width and height) of the feature maps while retaining important information. Common pooling operations include max pooling and average pooling. Max pooling is a specific type of pooling used here, after each convolutional layer with pool size of 2x2, the maximum value within each portion covered by the image is retained, highlighting the most dominant features. This helps in reduce the spatial dimensions of the feature maps, helping in reducing computational complexity and overfitting.

4. Flattening: Flattening converts the input into 1-D feature vector which are fed into the final layer of the CNN called the Dense or fully-connected layer.

5. Fully Connected Layer: There is one hidden dense layer with 128 neurons and ReLU (Rectified Linear Unit) activation used to add non-linearity. These layers perform classification or regression tasks based on the feature extracted from the flattened representation of the input.

6.Output Layer: The output layer consists of 'num_classes' neurons, where 'num_classes 'represents the number of classes in the classification task. SoftMax activation function is used to complete the probabilities of each class, ensuring that the output values are in the range [0,1] and sum up to 1

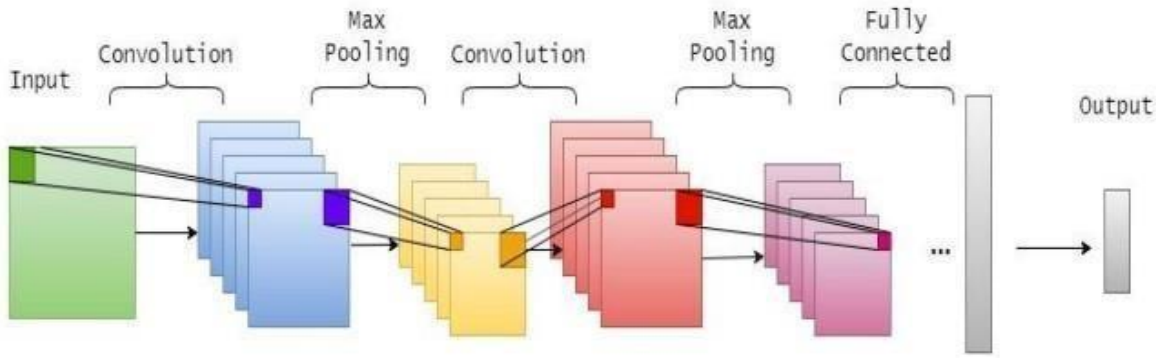

Figure 2 Block diagram of CNN model

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

Figure.2 depicts the block diagram of CNN. Overall, the proposed CNN model has two convolutional layers followed by max- pooling layers for feature extraction, a fully connected layer for high-level feature learning, and an output layer for classification. The SoftMax function is used by the model to classify input images into 'num_classes' categories to generate class probabilities.

The figure.3 illustrates the training process for Convolutional Neural Network (CNN) model used for image recognition. In the context of training a Convolutional Neural Network (CNN), the process begins with the training dataset, which undergoes preprocessing. Then the CNN model is subsequently built. The CNN model is then fed the gray scaled images that were acquired during preprocessing. Next, the images are classified by the trained CNN model. The classification step determines whether an image is correct or incorrect. Ultimately, the classification results are shown, letting the user know if the image was successfully classified or not.

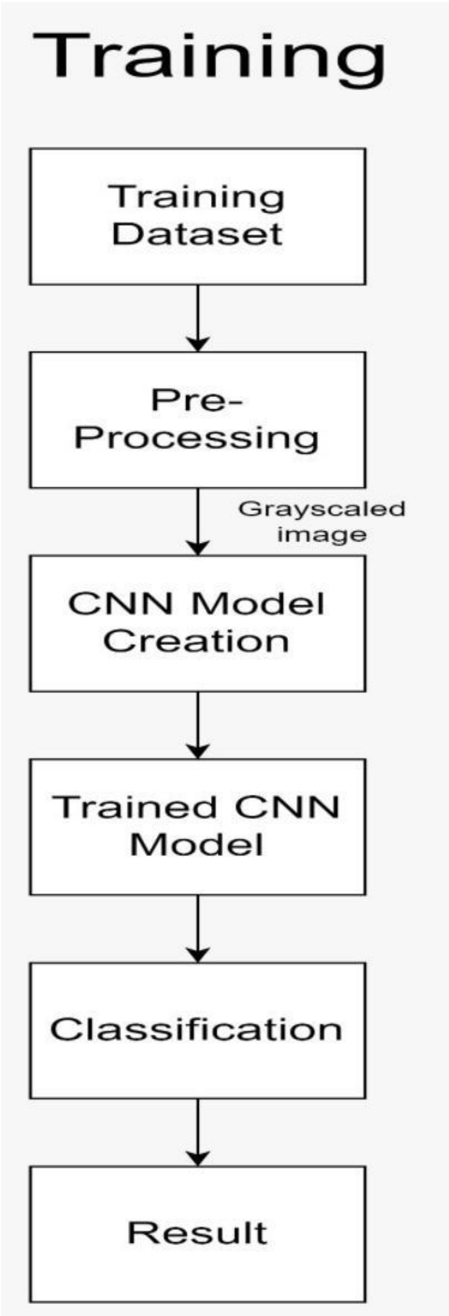

Figure 3 Methodology for Training Phase

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⧘ Peer-reviewed & Refereed journal ⧘ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

The testing process for the Convolutional Neural Network (CNN) model is shown in Figure 4. The process begins with the testing dataset, which undergoes preprocessing. The Gray scaled images that were acquired during preprocessing is now fed into the trained CNN model. Next, the images are classified by the trained CNN model. The classification step determines whether an image is correct or incorrect. Ultimately, the classification results are shown, letting the user know if the image was successfully classified or not.

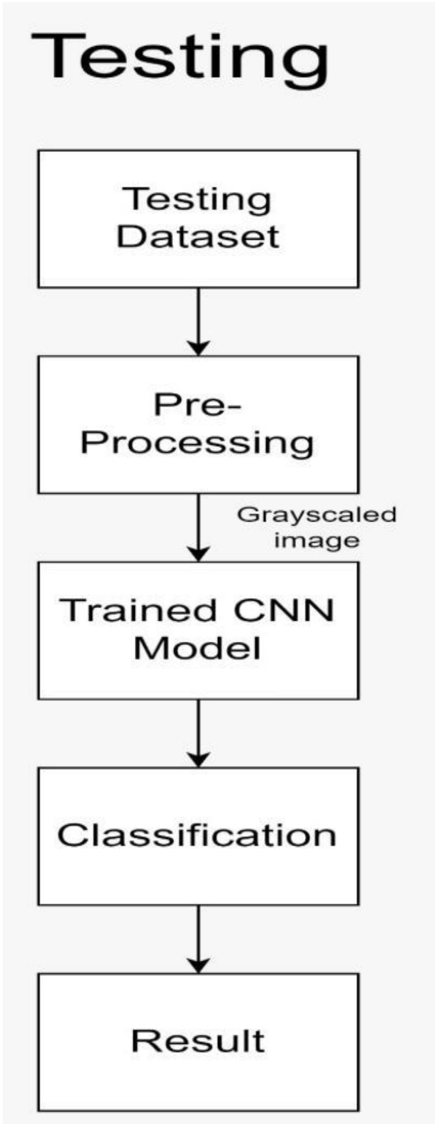

Figure 4 Methodology for Testing Phase

IV. RESULTS AND DISCUSSION

The project "Handwritten Character Recognition of Kannada Language using Convolutional Neural Network" was carried as an experiment to train the model to detect the characters whichever feed as the input and to get the specified output. There are many other machine learning algorithms which can be implement for this particular project.

Particularly CNN is the best method to implement compared to all the other algorithms. Initially, the datasets were split into a training and testing datasets. The dataset consists of 16,425 images in which 70% is used for training phase and remaining 30% is used for testing phase. Impressively, the model achieved the accuracy rate of 97% during the testing phase. This split ensures that the models are tested on untested data to evaluate their generalization skills after being trained on a subset of the data.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024
DOI: 10.17148/IJARCCE.2024.13554

Further research is showed that the Convolutional Neural Network performed greater in predicting the characters whichever is given as the input. Convolutional Neural Network are a class of deep neural network designed for tasks such as image recognition and computer vision. Here it consists one input layer, three middle layers, one flatten layer, one dense layer and one output layer. Here each layer consists of neurons.

The CNN algorithm is used in this research to identify handwritten Kannada characters. Typically, a CNN is made up of multiple layers that cooperate to extract hierarchical characteristics from the input data. The CNN's architecture is usually formed by stacking these layers one after the other. While the fully connected layers and output layers are in charge of feature extraction. The uploaded image is per-processed to grayscale in order to reduce disruption. The grayscale image conversion makes the data easier to understand for neural networks But its important to note that the model whichever is designed is limited to recognizing characters within the specified datasets. As a result, when inputted with characters beyond these sets, it generates a blank output.

The result of this experiment emphasizes with the collection of training data, which is large set of images with their corresponding classification. Before the CNN can learn from this data, the images might undergo per-processing steps like conversion to grayscale to ensure consistency. The core of the process is the CNN model itself. This neural network is like a complex web of connections that learns to identify patterns within the training images. By feeding the per-processed images through the CNN model, the network adjusts its internal connections to better recognize the features that differentiate between different classifications. This iterative process of feeding data and adjusting connections is what trains the model. The ultimate goal is to create a "trained CNN model" that can take a completely new image as input and accurately classify it based on the patterns it learned it learned from the training data. The result, displayed at the end, would be the classification label assigned by the trained model to the new image.

Furthermore, in the context of testing a Convolutional Neural Network (CNN), the process begins with the testing dataset, which undergoes per-processing. The Gray scaled images that were acquired during per-processing is now fed into the trained CNN model. Next, the images are classified by the trained CNN model. The classification step determines whether an image is correct or incorrect. Ultimately, the classification results are shown, letting the user know if the image was successfully classified or not. With the focus on preserving historical records in the state, this project addresses the current need for efficient digitization techniques in the digital age.

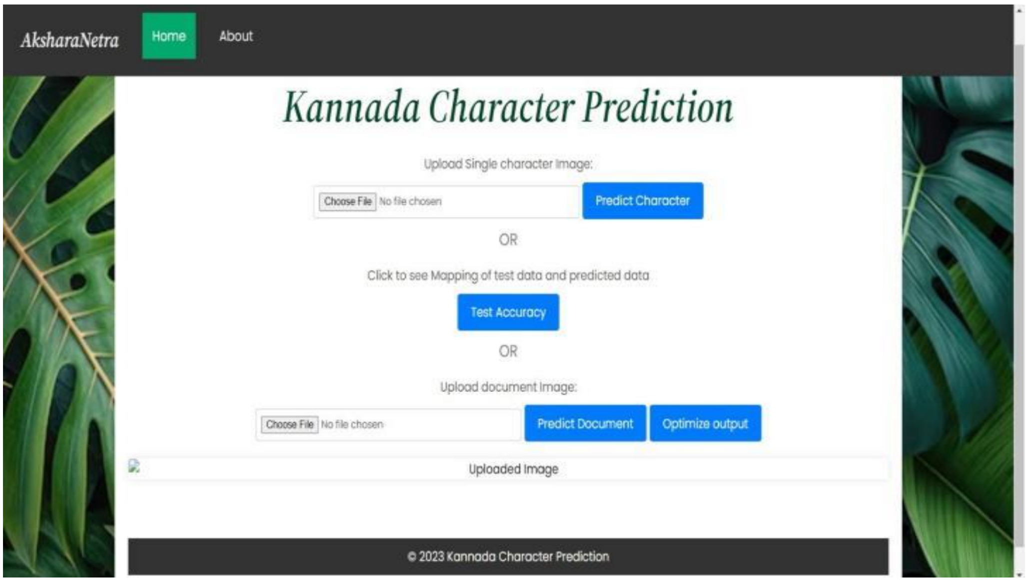

Figure 5 Home Page

The figure 5 describes the starting page of the project in which this has the options for uploading the images of character or sentence and predicted output can be seen by the user and also accuracy can be seen.

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 13, Issue 5, May 2024
DOI: 10.17148/IJARCCE.2024.13554

Figure 6 Upload character image

The figure 6 describes the character prediction for that particular character by uploading the single character image using the dataset provided for the user as an input by selecting choose file option in the window and that particular character can be predicted as an output or result.

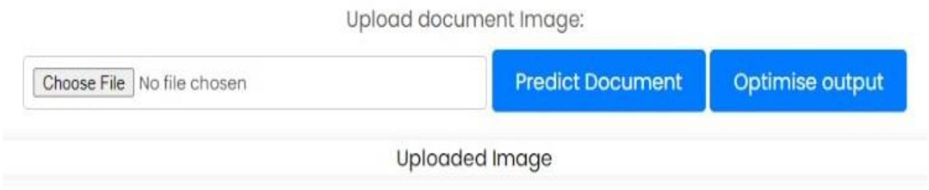

Figure 7 Upload Sentence image

The figure 7 describes the sentence prediction for that particular sentence by uploading the sentences images using the dataset provided for user as a input by selecting choose file option in the window and that particular sentence can be predicted as a output or result.

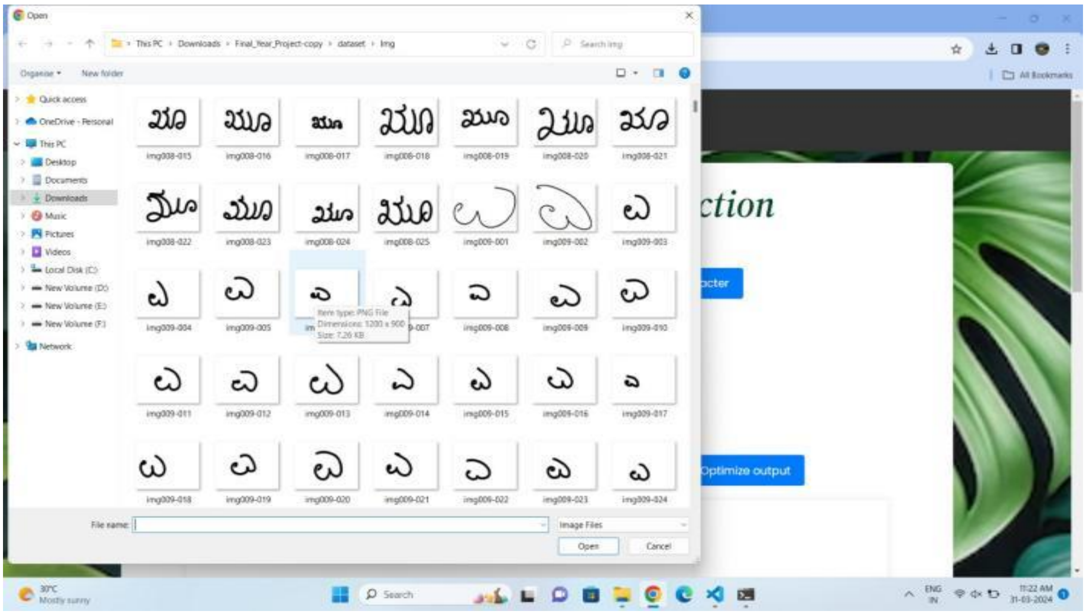

Figure 8 Uploading the character

The figure 8 describes the various types of characters present in the dataset. The user can select the images of these characters in the dataset provided and that particular character can be uploaded for prediction.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

Figure 9 Result of the given character

The figure 9 describes the result of the particular character that user has uploaded as the input from the given dataset and the result and confidence level is displayed.

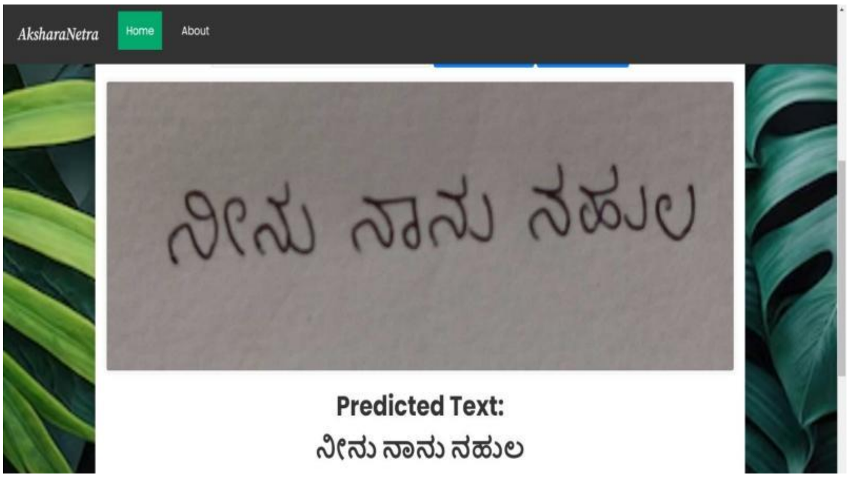

Figure 10 Result of the given Sentence

The figure 10 describes the result of the particular sentence which has been uploaded by the user from the given dataset to see the result.

Table 1: Table of number of character and its respective accuracy and time taken for prediction

Number of Character	Accuracy	Time taken for Prediction
10	100.00%	1.58 seconds
45	95.56%	7.24 seconds
5000	96.53%	447.85 seconds

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13554

The table 1 describes that number of character and their respective accuracy and time taken for prediction. The number of characters is taken like 10 and the accuracy for 10 characters will be 100% and the time taken for the prediction is 1.58 seconds.

```
Mapping of Testing Data and Predicted Data:
 Actual Class Predicted Class  Prediction Time (seconds)
          412           [412]                   0.663596
           56            [56]                   0.152111
          180           [180]                   0.158606
          109           [109]                   0.145660
          548           [548]                   0.134317
          511           [511]                   0.138824
          232           [232]                   0.143632
          405           [405]                   0.140142
          363           [363]                   0.142435
          280           [280]                   0.143162
          186           [186]                   0.131238
          412           [412]                   0.137780
          459           [459]                   0.137220
          179           [179]                   0.157030
          252           [252]                   0.138272
          175           [175]                   0.148746
          168           [168]                   0.144279
          235           [235]                   0.146861
          521           [521]                   0.150460
          438           [438]                   0.149621
          644           [644]                   0.147668
           83            [83]                   0.136261
          248           [248]                   0.136096
          482           [482]                   0.150686
          626           [626]                   0.137794
          492           [492]                   0.139848
          205           [205]                   0.085608
          543           [543]                   0.089568
          238           [238]                   0.177270
          301           [301]                   0.152258
          167           [167]                   0.169254
           76            [76]                   0.178372
          340           [340]                   0.186472
          535           [535]                   0.162696
          516           [516]                   0.162781
          538           [538]                   0.167925
          273           [273]                   0.192994
          189           [189]                   0.210268
          100           [100]                   0.185887
          408           [408]                   0.151872
          151           [151]                   0.147396
          592           [592]                   0.148308
          546           [546]                   0.138758
          412           [423]                   0.140278
          229           [619]                   0.141899

Testing Accuracy: 95.56%
Total time taken for predictions: 7.24 seconds
```

Figure 12 Testing Accuracy for random 45 Characters

The figure 12 describes the accuracy rate for random 45 characters and gives the total time taken for prediction and also the Testing Accuracy of 95.56%.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024
DOI: 10.17148/IJARCCE.2024.13554

Table 2: Table showing the Accuracy rate and time of prediction of each character

Character	Accuracy	Time of Prediction
ಅ	100.00%	1.58 seconds
ಆ	100.00%	1.82 seconds
ಇ	100.00%	1.82 seconds
ಈ	100.00%	1.60 seconds
ಉ	88.00%	1.95 seconds
ಊ	100.00%	2.02 seconds
ಋ	88.00%	1.72 seconds
ೠ	96.00%	1.70 seconds
ಎ	100.00%	1.50 seconds
ಏ	100.00%	1.52 seconds
ಐ	100.00%	2.73 seconds
ಒ	96.00%	1.70 seconds
ಓ	84.00%	1.65 seconds
ಔ	96.00%	1.56 seconds
ಅಂ	100.00%	1.51 seconds
ಅಃ	92.00%	1.91 seconds
೦	100.00%	1.66 seconds
೧	100.00%	1.60 seconds
೨	100.00%	1.58 seconds
೩	96.00%	1.68 seconds
೪	88.00%	1.78 seconds
೫	96.00%	1.71 seconds
೬	100.00%	1.59 seconds
೭	100.00%	1.71 seconds
೮	100.00%	1.71 seconds
೯	100.00%	1.71 seconds

The Table 2 describes the Accuracy rate time taken for the prediction of each character and different accuracy and time taken for prediction has been obtained for the characters.

V. CONCLUSION

The development of "Handwritten Character Recognition of Kannada Language Using CNN" project utilizes a deep learning technique called Convolutional Neural Network (CNN) to tackle the problem of converting the handwritten character to digital format. Convolutional Neural Network (CNN), has effectively extracted the features from the input image which helped the model in better classification of Kannada handwritten character recognition.

We can put the method into practice and test the model to determine its accuracy by collecting and utilizing data. The character can be recognized with a certain accuracy by the model.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⅀ Peer-reviewed & Refereed journal ⅀ Vol. 13, Issue 5, May 2024
DOI: 10.17148/IJARCCE.2024.13554

ACKNOWLEDGMENT

We would like to express our profound gratitude to Canara Engineering College for their constant encouragement and assistance during the preparation and presentation of our paper. Furthermore, we would like to sincerely thank **Dr. Sunil Kumar B L**, our guide, whose insightful advice and knowledge have been important in structuring our work and improving our academic careers.

REFERENCES

[1]. A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey and P. Tiwari, "Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network," in *IEEE Access*, vol. 7, pp. 7717-7727, 2019, doi: 10.1109/ACCESS.2018.2888882.

[2]. S. Afroge, B. Ahmed and F. Mahmud, "Optical character recognition using back propagation neural network," *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, Rajshahi, Bangladesh, 2016, pp. 1-4, doi: 10.1109/ICECTE.2016.7879615.

[3]. P. Dhande and R. Kharat, "Recognition of cursive English handwritten characters," *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, Tirunelveli, India, 2017, pp. 199-203, doi: 10.1109/ICOEI.2017.8300915.

[4]. Z. Li, L. Jin and S. Lai, "Rotation-Free Online Handwritten Chinese Character Recognition using Two-Stage Convolutional Neural Network," *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 205-210, doi: 10.1109/ICFHR-2018.2018.00044.

[5]. R. Fernandes and A. P. Rodrigues, "Kannada Handwritten Script Recognition using Machine Learning Techniques," *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Manipal, India, 2019, pp. 1-6, doi: 10.1109/DISCOVER47552.2019.9008097.

[6]. T. C. Wei, U. U. Sheikh and A. A. -H. A. Rahman, "Improved optical character recognition with deep neural network," *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, Penang, Malaysia, 2018, pp. 245-249, doi: 10.1109/CSPA.2018.8368720

[7]. Zhe Li, Lianwen Jin, Songxuan Lai"Rotation-Free Online recognition of handwritten Chinese characters utilizing two-stage convolution neural network."12th International Conference on Computing Communication and Networking Technologies (ICCCNT).

[8]. S. Hassan, A. Irfan, A. Mirza and I. Siddiqi, "Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting," *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, Istanbul, Turkey, 2019, pp. 67-72, doi: 10.1109/Deep-ML.2019.00021.

[9]. Afroge, S., Ahmed, B., & Mahmud, F"Optical character recognitionusing back propagation neural network." 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE) (pp. 1-4). (IEEE).

[10]. N. Majid and E. H. Barney Smith, "Introducing the Boise State Bangla Handwriting Dataset and an Efficient Offline Recognizer of Isolated Bangla Characters," *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 380-385, doi: 10.1109/ICFHR-2018.2018.00073.

[11]. M. K. Mahto, K. Bhatia and R. K. Sharma, "Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition," *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015, pp. 59-65, doi: 10.1109/ICACEA.2015.7164646.

[12]. Rao, Abhishek S., et al. "Exploring deep learning techniques for Kannada handwritten character recognition: a boon for digitization." *International Journal of Advanced Science and Technology* 29.5 (2020): 11078-11093.

[13]. Hallur, Vishweshwrayya C., and R. S. Hegadi. "Handwritten Kannada numerals recognition using deep learning convolution neural network (DCNN) classifier." *CSI Transactions on ICT* 8 (2020): 295-309.

[14]. U. Pal, N. Sharma, T. Wakabayashi and F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts," *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Curitiba, Brazil, 2007, pp. 749-753, doi: 10.1109/ICDAR.2007.4377015

[15]. N. Majid and E. H. Barney Smith, "Introducing the Boise State Bangla Handwriting Dataset and an Efficient Offline Recognizer of Isolated Bangla Characters," *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 380-385, doi: 10.1109/ICFHR-2018.2018.00073.

