Received: 2 January 2023 | Revised: 17 March 2023 | Accepted: 22 March 2023 | Published online: 3 April 2023

RESEARCH ARTICLE

Artificial Intelligence and Applications
2023, Vol. 1(3) 179–190
DOI: 10.47852/bonviewAIA3202624

BON VIEW PUBLISHING

Comprehensive Dataset Building and Recognition of Isolated Handwritten Kannada Characters Using Machine Learning Models

Chandravva Hebbi[1,2,*] and H. R. Mamatha[1]

[1]*Department of Computer Science and Engineering, People's Education Society University, India*

[2]*Visvesvaraya Technological University, India*

Abstract: In this work, an attempt is made to build a dataset for handwritten Kannada characters and also to recognize the isolated Kannada vowels, consonants, modifiers, and ottaksharas. The dataset is collected from 500 writers of varying ages, gender, qualification, and profession. This dataset will be used to recognize the handwritten kagunitas, ottaksharas, and other base characters, where the existing works have addressed very less on the recognition of kagunitas and ottaksharas. There are no datasets for the same. Hence, a dataset for handwritten 85 characters is built using an unsupervised machine learning technique, that is, K-means hierarchical clustering with run length count features. An accuracy of 80% was achieved with the unsupervised method. The dataset consists of 130,981 samples for 85 classes; these classes are further divided into upper, lower, and middle zones based on the position of the character in the dialect. After the dataset was built, support vector machine model with histogram of oriented gradients features was used for recognition and an accuracy of 99.0%, 88.6%, and 92.2% was obtained for the upper, middle, and lower zones, respectively, to increase the recognition rate, the convolutional neural network model is fine-tuned with raw input, and an accuracy of 100%, 96.15%, and 95.38% was obtained for the upper, middle, and lower zones, respectively. With the ResNet18 model, an accuracy of 99.88%, 98.92, and 97.55% was obtained for each of the zones, respectively. The dataset will be made available online for the researchers to carry out their research on handwritten characters, kagunitas, and word recognition with segmentation.

Keywords: run length count, K-means hierarchical clustering, convolutional neural network, support vector machine (SVM), histogram of oriented gradients

This work is aimed to build a dataset for handwritten Kannada characters and recognize these characters using the machine learning models.

1. Introduction

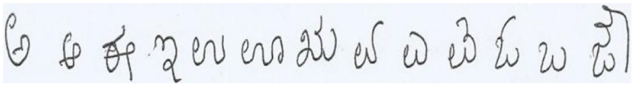
Technological developments have made computers read, process, and understand images. The images can be document images, natural scene images, or photographs. The document images can be printed or handwritten, and the sources from which these documents are generated include government offices, old textbooks, notes, form filling, etc. To store these documents in the systems, images require larger storage space. To reduce the storage space required, these documents can be digitized and the information can be maintained in digital form. The data present in digital form can be easily retrieved and processed for later use. To digitize these documents, various machine learning models and feature extraction methods can be applied using the benchmark datasets. Benchmark datasets for handwritten characters are available for the languages like English the MNIST, CEDAR, and CENPARMI datasets, Arabic, Chinese, Japanese, and Korean Scripts (Shailesh & Prashnna, 2016). Some of the Indic script datasets such as Devanagari (Mohammed et al., 2017), Bangla the CMATERDB, ISI, BanglaLekha Isolated, and Oriya, Telugu (Velpuru et al., 2020), and Tamil (Dave, 2021) are made publicly available. For some of the other south Indian scripts like Kannada, the datasets made publicly available do not include all the characters present in the character set (Bellary & Kusumika, 2020) and the number of samples per class is very less. Hence, the researchers have used their own datasets for the research work and have considered a subset of characters. This motivated us to work on building the benchmark dataset for handwritten Kannada characters viz vowels, consonants, modifiers, and ottaksharas, and make these data publicly available for all the researchers. Various image processing and machine learning techniques are applied to build the dataset and recognize the characters. This dataset helps to recognize the handwritten base characters, kagunitas, and ottaksharas with a minimal number of classes.

*Corresponding author: Chandravva Hebbi, Department of Computer Science and Engineering, People's Education Society University & Visvesvaraya Technological University, India. Email: chandravvahebbi@pes.edu

About the Kannada script: Kannada script is one of the Dravidian languages of south India mainly in the Karnataka state. It is used to write Kannada which is the official language of the Karnataka state. The character set consists of 34 vyanjanagalu (consonants), 13 swaragalu (vowels), and 2 yogavahagalu (Kannada script, 2023). The combination of consonants and swaragalu forms the gunitaksharagalu (kagunitas), which are 510 in number. Letters representing consonants are combined to form digraphs (ottakshara) when there is no intervening vowel. Each character has its own structure and sound giving visible and audible representations.

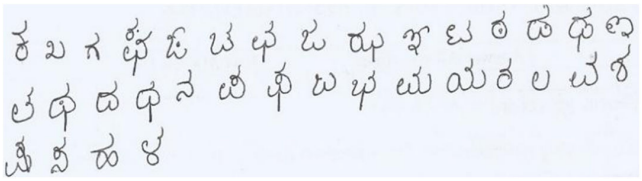
Kannada vowels: The Kannada character set consists of 13 vowels as shown in Figure 1.

Figure 1
Kannada vowels



Consonants: The character set consists of 34 consonants as shown in Figure 2.

Figure 2
Kannada consonants



Modifiers: There are 11 modifiers as shown in Figure 3.

Figure 3
Handwritten Kannada modifiers

Ottaksharas: The Kannada character set consists of 35 characters in ottakṣhara as shown in Figure 4(a). Figure 4(b) shows the combination of consonant with consonant to form ottakṣharas.

Kagunitas: Kagunitas are the characters that are obtained by the combination of consonants with vowels. There are 34 consonants and 13 vowels, so there are 510 kagunitas characters present in Kannada. Sample kagunitas are shown in Figure 5.

The work in the field of handwritten characters namely consonants and vowels is carried out extensively. Some researchers have worked on the kagunitas, which has 510 classes. In such a case, the number of classes will be more than 600+ considering the vowels, consonants, ottaksharas, and digits. The number of classes is reduced from 600+ to 93 classes (excluding digits), the recognition rate is increased, and the training time is reduced. The proposed method works with minimum classes to recognize the handwritten Kannada characters.

In this research, an exhaustive dataset for the handwritten Kannada 85 characters is collected from a heterogeneous group of people. The rest of the paper is organized into a literature review, methodology to build a dataset, results, and discussion. The rest of the work is organized into a literature survey, proposed methodology, results and discussions, and a conclusion.

2. Literature Review

The work carried out in the field of handwritten characters dataset building and recognition of these characters is presented. Part A of the literature review is on the dataset building and part B is on the recognition of handwritten Kannada characters.

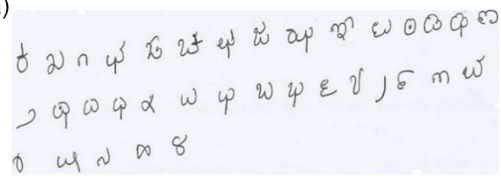
2.1. Dataset building

The multilingual Turkish-Hungarian-English (T-H-E) dataset development and recognition of characters are presented by Bartos et al. (2020). LeNet, a deep learning model, is used to confirm the validity of the dataset. All the images considered were of size 32 × 32. The dataset consists of 52 classes written by 200 participants. Augmentation techniques are applied to increase the size of the dataset and increase the performance of the model. Very good accuracy was reported for different datasets created. Lee et al. (2021) have built a dataset for English, Good Notes Handwriting Kollection; the dataset comprises 172,936 characters, 39,026 words, mathematical expressions, ASCII symbols, and 9363 lines. Mask R-convolutional neural network (CNN) is used for segmentation, and Clova AI deep is used for recognition. The authors have claimed an accuracy of 86.1%.

Musa (2011) presented the SUST-ALT (Sudan University of Science and Technology-Arabic Language Technology group) research group that has developed an Arabic handwritten offline characters and numerals dataset. The support vector machine (SVM) classifier was used to recognize numerals and multi-layer perceptron for the alphabet. Very good accuracy was reported by the researchers. Alamri et al. (2008) have discussed methods to build a dataset for handwritten Arabic dates, isolated letters, words, numerical strings,

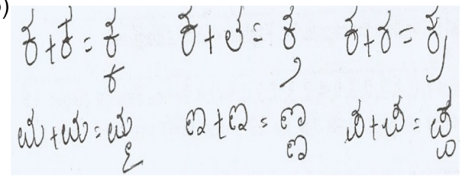
Figure 4
Kannada ottakṣhara characters

(a)



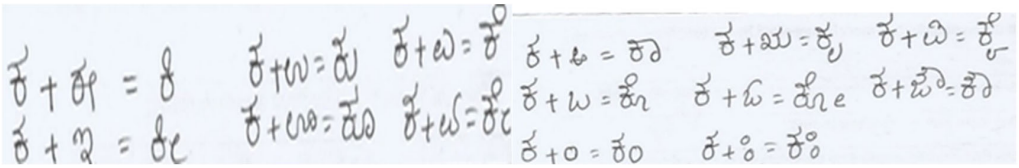
Ottaksharas

(b)



Consonant-Consonant Combination

Figure 5
Kannada handwritten kagunitas



and special symbols. Pre-processing methods are used for noise removal and binarization. Segmentation was applied to extract the individual letters or characters. K-nearest neighbor (K-NN) classifier with $K = 3$ is used for classification. Three datasets are built with varying number of samples. Very good accuracy was reported.

Sagheer et al. (2009) have presented methods to build and recognize handwritten Urdu-isolated characters, digits, special symbols, words, and numeral strings. The SVM classifiers with gradient features were used for the recognition of digits. An accuracy of 98.61% was claimed for handwritten digits. Ali et al. (2020) have described a technique to build a dataset for handwritten Urdu characters and digits. The dataset comprises digits and characters collected from 900+ individuals. Autoencoder neural network and CNN models were used for the recognition. Very good accuracy was claimed.

Ahmed et al. (2021) have discussed a mechanism to build the dataset for handwritten Kurdish characters. The dataset consists of samples written by 390 writers. There are 35 classes with each class consisting of 1134 samples. Accuracy was not mentioned by the researcher. Dataset building for Pashto, a national language of Afghanistan, is presented by Uddin et al. (2021). Data are collected from 350 college students with ages 19–24 years. The number of characters in the dataset is 4300 characters for 43 classes. Augmentation techniques are applied to increase the size of the data. The deep neural network (DNN)-based backpropagation algorithm with Rectified Linear Unit (ReLU) activation function for the classification of Pashto handwritten characters was used to evaluate the performance of the dataset. The accuracy of the model reported is 87.67% with one activation ReLU layer.

Kavallieratou et al. (2001) discussed the methods to recognize handwritten Greek numerals and characters. The dataset consists of 667,583 symbols and 102,692 words collected from 1000 writers. How the data were moved to the bins is not specified and also the accuracy of the model is not mentioned. Margaronis et al. (2009) have developed a dataset for handwritten Greek characters, words, and symbols. The dataset consists of 24 Greek letters, 25 small letters, 10 numerals, and 6 punctuation symbols. The dataset consists of 428 filled forms written by 305 unique persons. The dataset consists of 39,823 letters, 8426 digits, 12,519 images of words, and 5696 others including special characters and punctuation marks. Liu et al. (2011) have discussed the technique to build the dataset for Chinese handwritten characters. The dataset is built for both online and offline methods. The dataset comprises 3.9 million samples for 7356 classes. Two linear discriminant analysis classifiers are used to separate printed and handwritten characters. The projection profiles are used for character segmentation. The connected component clustering-based algorithm is used for the annotation of the characters. The accuracy of the method is not mentioned.

Shahariar Azad Rabby et al. (2018) have presented the dataset building for handwritten Bangla handwritten vowels, consonants, modifiers, compound letters, and numerals. Pre-processing techniques like binarization, thresholding, and noise removal are applied to collect data. The dataset comprises 367,018 samples for 122 classes written by 3086 persons. An accuracy of 96.90% was claimed. Ferdous et al. (2021) have presented a method to build a dataset for handwritten Bangla compound characters. Pre-processing techniques like thresholding and noise removal are applied to the scanned document. The dataset built consists of 306,240 samples for 120 classes, written by 2552 writers. Biswas et al. (2017) have presented a method to develop a dataset for handwritten Bangla characters. The dataset consists of 1,66,105 samples for 84 classes collected from 2000 writers. The pre-processing techniques applied are binary inversion, noise removal, and edge thickening.

Das et al. (2014) have presented the technique to build the dataset for handwritten Bangla compound characters. The dataset comprises 55,278 samples for 199 classes of compound characters. Only 171 classes are used for the experimentation. The SVM classifier with convex hull and quadtree-based features is used to recognize the characters. An accuracy of 79.35% was claimed. Dongre & Mankar (2012) have described the dataset creation process for Devanagari handwritten characters. Data are collected from 750 people of varying ages, gender, and profession. Characters are segmented using horizontal projections and moved to their respective folder. The method used in moving the characters to their respective folders is not specified. The dataset consists of 5137 numerals and 20,305 characters.

A benchmark dataset building for handwritten Telugu characters is presented by Lakshmi et al. (2020). The data were collected from 200 writers. Pre-processing methods like binarization and noise removal are applied. After the pre-processing characters are extracted using the Otsu-threshold filter segmentation method, the dataset consists of 10,400 handwritten characters. The deep CNN model was used to evaluate the dataset built. An accuracy of 80% was claimed for 52 classes. Dataset building and recognition of Malayalam characters are presented by Manjusha et al. (2019). The samples are collected from 77 people aged between 20 and 60 years. Each of the characters is segmented using segmentation algorithms, and the characters are moved to their respective bins manually. The machine learning models like Lenet-5 and SVM with different evaluation metrics and feature extraction methods are experimented with a dataset to check the robustness of the algorithm. The dataset size reported is 29,302 characters for 85 classes. An accuracy of 91.05% was reported with scattering convolutional network.

2.2. Recognition of Kannada handwritten characters and numerals

Prabhu (2019) has presented the method to build the dataset for Kannada-MNIST numerals. The data are collected from 65 writers. Each writer was asked to write 128 instances of each digit. The handwritten digit document id was scanned, and various pre-processing methods like adaptive threshold, Gaussian blur, and

bitwise Not were applied. The dataset consists of 83,200 images. CNN model is used to evaluate the dataset. An accuracy of 97% was reported.

Dataset building for the handwritten Kannada vowels using the unsupervised learning method viz, K-means clustering followed by supervised learning method is discussed by Hebbi et al. (2021). The data were collected from 500 people of varying ages, gender, and profession. The dataset consists of 6500 for 13 classes. The performance of the dataset is tested by using feature extraction methods like local binary pattern, run length count (RLC), chain code, and histogram of oriented gradients (HOG), and the features are fed to the unsupervised machine learning models to create the class labels. The accuracy of the model was approximately 80%. The loss in accuracy was due to the presence of similar characters in the character set.

Aradhya et al. (2010) have presented the technique to recognize handwritten Kannada characters using a probabilistic neural network. The principal component analysis and Fourier transform are used for extracting features. The dataset consists of 500 samples for 50 classes. An accuracy of 68.89% was claimed. Authors have tested the proposed method on the COIL-20 object dataset, and an accuracy of 88.64% was reported. Parameshwarppa & Dhandra (2015) have discussed a method to recognize handwritten Kannada characters with the features from generation discrete curvelet transform (DCTG2) and a K-NN classifier. The dataset consists of 48 classes, 9600 characters with 200 samples per class. An accuracy of 92.21% was reported for both vowels and consonants.

Ramesh et al. (2019) have described the methods to recognize the handwritten Kannada characters using the capsule network. The dataset consists of 47 classes with 500 samples for each class. An accuracy of 98.7% was claimed. Rajput & Horakeri (2011) have presented the techniques to recognize handwritten Kannada vowels by using Crack Codes and Fourier descriptors features, and the use of K-NN and SVM classifiers. The dataset consists of 500 samples for each character. An accuracy of 91.24% and 93.73% was reported for K-NN and SVM classifiers, respectively. Authors have considered only the base characters for the work.

The methods to extract structural features like eccentricity, orientation, area, convex area, EquviDiameter, and perimeter are presented by Angadi & Angadi (2015). These features are input to the SVM classifier. The experiment was conducted on Kannada vowels and consonants with an accuracy of 89.84% and 85.14%, respectively, on a dataset consisting of 2490 samples for 49 classes out of 657+ classes.

Deep learning models to recognize handwritten Kannada characters are presented by Rao et al. (2020). The data are collected from Char74K, which consists of 657 classes with 25 samples per class. The pre-processing techniques like denoising, normalization, and binarization have been applied to the input image. The document is segmented into lines, lines to words, and words to individual characters using the bounding box method. Augmentation techniques are applied to increase the size of the dataset. The CNN model is trained and tested with samples present in the dataset. An accuracy of 86% was reported. The dataset does not consist of characters for ottaksharas. The number of samples per class is very less. The characters are hand-drawn and not handwritten. Here, authors have not considered complex characters and words for the recognition.

Kumar et al. (2020) have described the performance evaluation of classifiers used to recognize handwritten Gurumukhi characters. The classifiers considered are linear SVM, SVM-Radial Basis Function (RBF), random forest, Naıve Bayes, decision trees, and CNNs. The dataset consists of 7000 handwritten characters and 6000 numerals. An accuracy of 87.9% was reported for the random forest classifier.

Recognition of handwritten Kannada characters using the CNN model and transfer learning with VGG-16 is presented (Parikshith et al., 2021). The dataset considered is Char74K, which consists of 25 samples per class, and there are 657+ classes. Good accuracy was reported by the authors. In this work, the researchers have considered only 50 classes for experimentation. Augmentation techniques are used to increase the number of samples per class. Handwritten Kannada character recognition using capsule networks is discussed (Shobha Rani et al., 2022). The dataset comprises 7769 samples for 49 classes with samples collected from 200 writers. Good accuracy was claimed by the authors. The authors have not considered modifiers and ottaksharas for the work.

In Devaraj et al. (2022), the method to recognize handwritten Kannada characters which is presented using the CNN model is described. The Char74K dataset is considered for the work, which consists of 657+ classes, 25 samples for each class. Augmentation techniques are applied to increase the number of samples. An accuracy of 90% was reported. The Char74K dataset consists of hand-drawn characters using a tablet PC not handwritten characters.

Recognition for handwritten characters using the Kaggle dataset is presented (Chinmayee Bhat, 2022). The dataset consists of 657 classes with 25 samples per class. Augmentation techniques are used to increase the size of the dataset. The K-NN, SVM, and CNN methods are used for handwritten characters using the Kaggle dataset for handwritten Kannada characters. The dataset consists of handwritten vowels, consonants, kagunitas, some of the old Kannada characters, and numerals. Good accuracy was reported by the authors. The number of samples per class is very less and has not considered ottaksharas.

Recognition of handwritten Kannada words with a segmentation approach is discussed (Ravikumar & Sampathkumar, 2022). The dataset consists of 100+ classes with 75 samples per class. Augmentation techniques are applied to increase the number of samples per class. CNN model is used for classification. The model is tested with 1000 handwritten words. Each word is divided into characters and each character is recognized individually. The bounding box method is used to extract the characters. The authors have not considered all the classes. An accuracy of 96.12% was reported.

Classification and recognition of handwritten Kannada and English characters using graph edit distance are discussed (Roopa & Mahantesh 2022). The characters considered for the work are 52 English characters, 10 digits, and 35 Kannada consonant conjugates. An accuracy of 97.01 % was reported. The number of samples per class is not specified, and the authors have not considered vowels, modifiers, and ottaksharas.

From the literature survey, it is found that public datasets are available for English, Greek, Arabic, Chinese, Pashto, Urdu, and Kurdish scripts. For Indic scripts such as Bangla, Devanagari, and Telugu, public datasets are available. For the scripts like Gujarati, Malayalam, Odia, Tamil, and Kannada, public datasets are not available and the researchers have developed their own datasets. But when considered for Kannada, the dataset was built only for the vowels. It is also understood from the survey how to build the dataset from scratch. Hence, some of the methods specified have been adopted in building the dataset for the Kannada language. Most of the researchers have worked on a subset of characters mainly vowels and consonants. Some of the researchers have used the Char74K dataset for the work, which consists of hand-drawn characters and has only 25 samples per class. In the proposed method, the work is considered for the recognition of handwritten vowels, consonants, modifier characters, and ottaksharas. The dataset is collected from a heterogeneous group of people.

3. Proposed Methodology

The proposed methodology consists of two steps, data collection and dataset building, to evaluate the robustness of the dataset created using the machine learning models. Data are collected from a heterogeneous group of people. An unsupervised machine learning model is used to annotate the images instead of manually moving the images to their respective bins. After the image annotation, supervised and unsupervised machine learning models are used to validate the dataset built. The dataset built will be used to recognize the handwritten characters.

3.1. Data collection

The data are being collected from 500 people of varying ages, gender, educational qualification, profession, and mother tongue. The writer's information is collected to research writer identification in forensics. Figure 6 shows sample characters written by the writer.

Figure 6
Sample handwritten characters

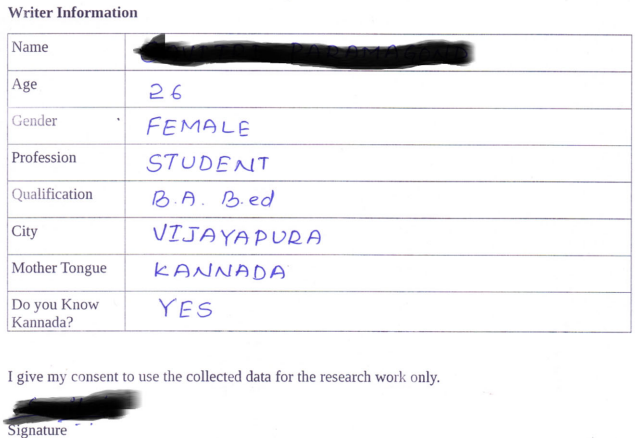
An A4 size sheet was given to each of the writers, and they are asked to write the characters within a grid. Each character is written only once. Writers had a restriction in writing the characters, namely characters have to be written within the grid and in sequence. Each of the writers was asked to write 84 characters, which include vowels, consonants, modifiers, and some of the ottaksharas available in the Kannada script.

3.2. Writers information

Figure 7 shows the same writer's information collected.

Table 1 also shows the age and gender of the writers. From Table 1, it is found that 38% of the data is collected from the people age group of 16–20 years. Seventy percent of the contributors for the dataset are students of engineering, BA, B.com, MA, MCA, B.sc, students of 4th standard to higher secondary education, and the remaining 30% from the people of other professions like housewives, bank employees, teachers, drivers, lawyer, doctor, etc.

Figure 7
Writer's information



Writer Information

Name	[redacted]
Age	26
Gender	FEMALE
Profession	STUDENT
Qualification	B.A. B.ed
City	VIJAYAPURA
Mother Tongue	KANNADA
Do you Know Kannada?	YES

I give my consent to use the collected data for the research work only.

[redacted]
Signature

Table 1
Writers' age and gender statistics

SI. No.	Age	Count	Male	Female
1	<=10	7	2	5
2	11–15	74	22	52
3	16–20	190	94	96
4	21–25	96	52	44
5	26–30	54	24	30
6	31–40	50	28	22
7	>=41	29	20	9

3.3. Dataset building

Each of the handwritten documents is assigned a unique integer value which is referred to as the writer's ID (identification number). All the documents are scanned using the Kyocera Ecosys FS-6525 MFP and HP Scanjet 200 Flatbed Scanner with 300 dpi as color images. Each of the scanned documents is saved in jpg image file format and named with the writer's ID. Figure 8 shows the proposed methodology to build the dataset and check its robustness using SVM and CNN classifiers.

3.2.1. Pre-processing

In the pre-processing step, all the document images with .jpeg are read. Each image is converted to a gray scale; followed by this, the image is inverted. The horizontal and vertical lines are removed from the document after applying the adaptive threshold. The other pre-processing methods applied are erosion, dilation, and noise removal using median blur. The characters are extracted from the document using the bounding box method. The extracted characters are saved to a common bin with the name of the file being the writer's ID followed by an integer number. For each writer, there were 88 characters approximately based on the segmentation. The total number of images in the bin is 44,273. Figure 9 shows the sample set of characters in the bin. The next step is a feature extraction

Figure 8
Proposed system architecture

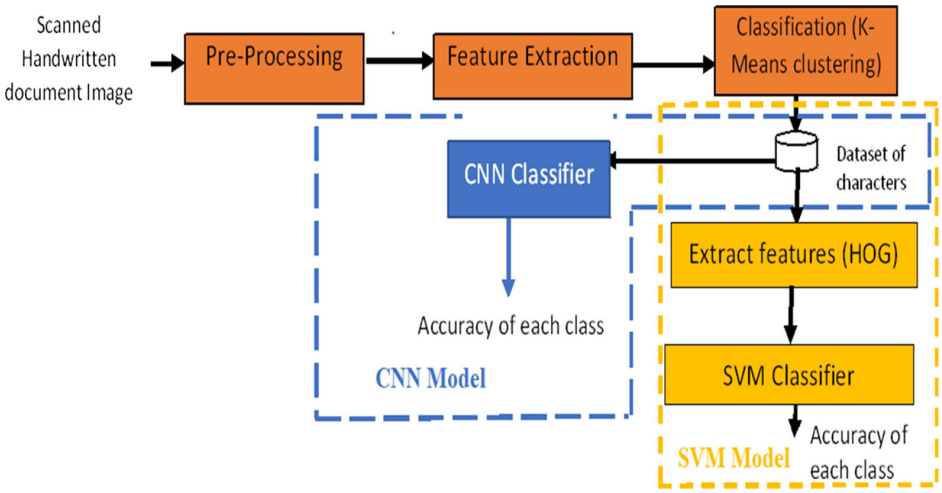
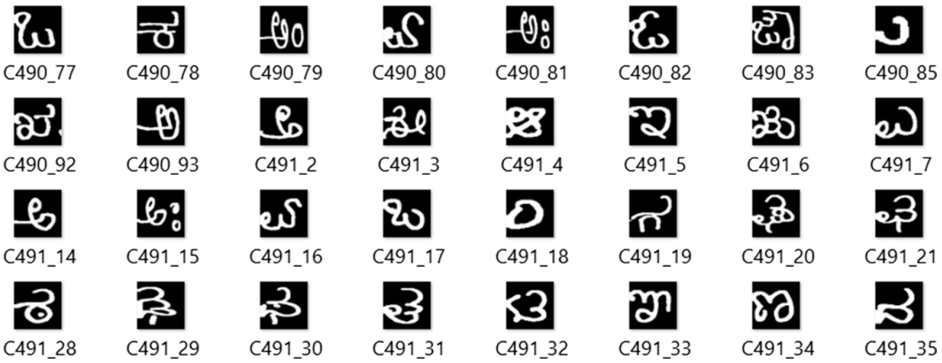


Figure 9
Sample characters after extraction from the image



3.2.2. Feature extraction

In this work, RLC features of an image are extracted. Run length encoding is one of the oldest, simple loss-less data compression algorithms primarily used to compress binary graphical data. The proposed algorithm for feature extraction is based on the number of edge shifts in the image. Figure 10 shows an example of the RLC whose block size is 5 × 5.

Input: 44,273 Handwritten Kannada character images.

Output: Vector with 100 extracted features.

Algorithm

Begin.

1. Read the images of size 50 × 50, then the image is binarized using Otsu's threshold method. 2. Divide the image into a block of 10 × 10 pixels. The total number of blocks will be 25.

3. For each block, the vertical and horizontal run length count is calculated. The number of features obtained is 25 × 4 = 100.

4. The numbers are normalized for each block in the image.

5. Normalized run length counts are taken as features of the clustering algorithms.

End. Hebbi et al. (2021).

Figure 10
Run length count for 5 × 5 size block.

1	0	1	1	0	**1, 2**
0	1	0	1	0	**2, 2**
1	1	1	1	0	**0, 1**
1	0	0	0	1	**1, 1**
0	0	1	0	1	**2, 1**
1, 2	**1, 1**	**2, 2**	**0, 1**	**1, 0**	***[5,6,6,7]***

3.2.3. Annotation using K-means clustering

All the 44,273 characters extracted from the A4 size sheet were moved to the single bin. These characters do not have the class labels to classify them using the supervised machine learning models. Hence, it was required to label these characters using machine learning models. An unsupervised K-means clustering algorithm with RLC features was used for the classification. Clustering can

help to identify and remove outliers or noise in the data, which can improve the quality of subsequent manual labeling. Additionally, clustering can be useful in identifying patterns or structures in the data. In terms of its impact on performance, the use of clustering can improve the efficiency and effectiveness of manual labeling by reducing the amount of redundant or irrelevant data that need to be labeled. This can save time and resources while still producing high-quality annotated data. Clustering is used to group data points based on similarity or distance metrics.

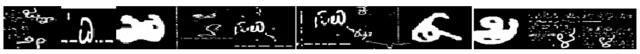
On average, there were 88 characters for each writer based on the segmentation. Hence, a K-value of 88 is being considered. The RLC features or RLC vectors of the images are extracted (as mentioned in Section 3.2.1) and are fed to the clustering algorithm for making clusters of similar characters. As Kannada has many characters to be with similar structures, all the similar characters were moved to the same bin.

After the classification, only 30% of the characters are correctly moved to the bins which are calculated manually. Hence, it was required to apply the clustering at level two with a K-value of 10. Thus, with the hierarchical clustering method, 80% of the characters were moved successfully to their bins. The remaining 20% of the characters are manually separated. The accuracy of the model is computed manually for each class. Challenges in classification are the similarity in the structure of the characters. After the classification, each of the bins had an unequal number of characters. The reason for the imbalance in the number of characters is the incorrect segmentation of the character; the character is wrongly written. The character written by the writer is (ka), but the character does not look like (ka). Similarly, the actual character is written as . Character is written as , character is written as , , , , and character is written as . The characters in Figure 11 have distortion at the bottom of the character as the characters were written within the grid; instead, the characters were on the grid line. During the grid removal, portions of characters were also removed. Such characters are not part of the dataset.

Figure 11
Characters trimmed at the bottom

In Figure 12, it is found that some of the characters had noise even after applying the noise removal method. Hence, these characters are excluded from the dataset.

Figure 12
Noisy and overwritten characters



Some of the characters are also wrongly segmented as indicated . Some of the writers have written the same character more than once and missed the other character. Based on these analyses, it is found that only 3% of the data is lost. With this, it is concluded that very good pre-processing steps were employed to segment the characters.

4. Results and Discussions

The results of the K-means clustering algorithm with RLC features, recognition of characters using SVM with HOG features, and recognition of characters with CNN have been presented.

4.1. Results of K-means clustering with RLC features

The challenges faced during the pre-processing were grids removal, the decision on structuring elements, and the number of iterations for which the dilation operation needs to be carried out to get the contours. After many trials, a good number of contours for the 2 × 3 structuring element and 15 iterations were obtained. Again, it was required to decide on the area of an image to be considered, that is, area of an image to be considered or rejected. If the image size is too large (i.e. height 350 pixels, width 320 pixels) are discarded. if the image size is too small (i.e. height and width <40 pixels) are discarded. After discarding the too small or too large area of the images, a single bin with 44,500 images was created. These image RLC features are fed as input to the K-means clustering algorithm for classification. The results of the K-means clustering algorithm were not good due to the presence of a similar structure of characters in Kannada. The characters are similar in structure and hence they were present in the same bin along with some of the other characters at level 1 of clustering. Hence, it is decided to apply the second level of clustering to the new clusters that were obtained. Based on the observation, the number of unique characters in each of the bins was approximately 10. There were 88 bins in total after level 2 clustering. From these bins, the data were moved to their respective bins manually. At the second level, 80% of the characters were correctly moved to their bins. Later these bins are classified into the upper zone, the lower zone, and the middle zone. The table shows the statistics of the dataset.

Table 2
Statistics of the number of classes and the total number of samples per class

SI. no.	Character type	No. of classes	No. of characters
1	Vowels	13	6702
2	Consonants	34	16,907
3	Modifiers	11	6181
4	Ottaksharas	25	11,761
5	Yogavahagalu	2	610

The number of unique classes is 85. For our experimentation, Yogavahagalu has been excluded. Hence, the dataset consists of 83 unique classes and is named Kannada83_1. Each of the classes does not have the same number of samples. This imbalance in the number of samples may hamper the performance of the system. Hence, it is decided to balance the samples per class. There are different methods to balance the dataset. *Method 1: Oversampling*. In this method, the classes with a maximum number of samples were found. Then samples of other classes are also increased to this value. Here, it was found that some of the classes had 750 samples. Hence, the samples per class were increased by applying rotation at an angle of -10°. From this, dataset 2 was obtained and it is named Kannada83_2. *Method 2: Undersampling*. In this

method, the samples are selected based on the minimum number of samples in the class. It was found that one of the classes had 319 samples; hence, 319 samples were randomly selected from each class. From these samples, dataset 3 was obtained and this dataset is named Kannada83_3. The 83 classes are divided into upper zone characters, middle zone characters, and lower zone characters based on the position of the character. Table 3 presents statistics for the number of zones, classes per zone, and the number of samples in each dataset. To check for the robustness of the dataset built, each of the datasets is being experimented with the HOG features of the images and SVM classifier, and CNN model.

Table 3
Statistics of the number of zones, classes per zone, and the number of samples in each dataset

			Total number of characters		
SI. no.	Zone	# Classes	Dataset 1	Dataset 2	Dataset 3
1	Upper zone	4	2228	3000	1276
2	Middle zone	53	27,253	39,750	16,907
3	Lower zone	26	12,159	19,500	8294
Total number of characters			41,644	62,250	26,477

To check for the robustness of the dataset built, each of the datasets is being experimented with an SVM classifier using the HOG features of the characters, CNN model, and ResNet18 model.

4.2. Image recognition with SVM and HOG features

4.2.1. Feature extraction

A HOG is a feature descriptor that is used to extract features from image data. The HOG feature extraction method is based on the direction and gradient features of the image. HOG with orientations = 9, pixels per cell = (8,8), cells per block = (4,4) are considered. Along with these, block normalization is done with L2 Hys (Hys stands for hysteresis). The output of this method is a one-dimensional array of feature descriptors. These features are fed into the SVM classifier.

4.2.2. Recognition model

The datasets have been experimented with the HOG feature descriptors and SVM classifier. Tables 4, 5, and 6 present the results of the SVM classifier with varying values of gamma, C—the regularization function, and kernel functions. Considering gamma= [1,0.1,0.02,0.002,.2], kernel = "rbf", probability = True, and varying the C value from 1 to 200, it was observed that increasing the C value up to 200 accuracies of the model increased gradually. When the C value was greater than 200, it was found that the accuracy of the model started decreasing; hence, the C value of 200 is considered. These parameters are considered for Kannada83_1, Kannada83_2, and Kannada83_3. It was found that the accuracy of the model was good with dataset 2. The results of dataset 2 for the middle zone, lower zone, and upper zone are presented in Tables 4, 5 and 6.

Table 4
Performance of dataset 2 with middle characters

		RBF with gamma						
SI. no.	C	1	0.1	0.02	0.001	0.2	Linear	Sigmoid
1	1	30.2	87.2	83.3	70.5	84.3	83.3	76.2
2	10	39.9	88.6	87.7	80.8	84.9	82.9	68.4
3	100	39.9	88.6	87.8	84.2	84.9	82.9	64.9
4	200	39.9	88.6	87.8	84.0	84.9	82.9	64.5

Table 5
Performance of dataset 2 with lower zone characters

		RBF with gamma						
SI. no.	C	1	0.1	0.02	0.001	0.2	Linear	Sigmoid
1	1	36.9	90.7	87.5	77.2	87.1	87.7	81.5
2	10	49.7	92.2	91.3	85.0	89.7	87.6	75.6
3	100	49.7	92.2	91.5	88.7	89.7	87.6	72.3
4	200	49.7	92.2	91.4	88.4	90.0	87.6	72.1

Table 6
Performance of dataset 2 with upper zone characters

		RBF with gamma						
SI. no.	C	1	0.1	0.02	0.001	0.2	Linear	Sigmoid
1.	1	73.0	98.8	98.6	96.1	98.2	98.8	97.5
2.	10	73.3	99.0	99.0	98.1	98.3	98.8	96.2
3.	100	73.3	99.0	99.0	98.7	98.3	98.8	94.8
4.	200	73.3	99.0	99.0	98.8	98.3	98.8	94.3

Considering the value of $C = 10$, $\gamma = 0.1$, and the RBF kernel, the performance of the datasets is presented in Table 7. Dataset 2 performed better compared to the other two datasets. In the case of middle zone characters, the accuracy of the dataset was 88.6%. The accuracy of middle zone characters was less compared to upper and lower zone characters due to the presence of more classes in the middle zone. Hence, for further experimentation with other machine learning models, dataset 2 was considered. The accuracy of the SVM model for linear and sigmoid was less compared to SVM with RBF kernel function.

Table 7
Comparison of performance of the datasets

Zone	Dataset 1	Dataset 2	Dataset 3
Upper zone	98.3	99.0	98.7
Middle zone	85.1	88.6	83.4
Lower zone	88.3	92.2	86.7

From Table 7, it is found that the accuracy of dataset 2 is good compared to the other two datasets. For the upper zone characters, there was not much change in the accuracy for 3 datasets as the number of classes was less and the chances of misclassification were less. In dataset 2, the accuracy of middle zone characters is less compared to the upper zone and lower zone as the number of cases is 53 in the middle zone.

4.3 Recognition with CNN model

The CNN model is trained with handwritten characters, and the 80–20 rule is used for training and testing. The model experimented with the dataset created and the performance of

Table 8
Accuracy of CNN model

	Dataset 1		Dataset 2		Dataset 3	
	#Epochs	Accuracy %	#Epochs	Accuracy %	#Epochs	Accuracy %
Upper zone	200	99.77	300	100	200	99.21
Middle zone	200	93.70	300	96.15	300	92.31
Lower zone	300	95.71	300	95.38	100	91.13

the model is evaluated based on max pooling and average pooling. The activation functions like tanh, ReLu, and SoftMax layer have been considered.

The impact of using max pooling, average pooling, activation functions, batch size, number of epochs, kernel initializers, and number of neurons in the dense layer have been analyzed. The model is fine-tuned for the parameters mentioned. The other kernel initializer functions considered are he_uniform, he_normal, random_uniform, and random_normal. Hence, the he_normal kernel function was considered. The number of epochs considered for training the model is 10, 20, 100, 200, and 300. Batch sizes were 64, 128, 256, 512, and 1024. Learning rate was 0.001, 0.01, 0.1, 0.2, and 0.3. Activation functions considered were softmax, softplus, softsign, relu, tanh, sigmoid, hard sigmoid, and linear. When the number of epochs was increased from 300 to 400 and other higher values, the accuracy of the model did not increase and sometimes there was even a drop in the accuracy. Hence, it has been decided the number of epochs is 300. There was not much change in the accuracy of the model when the batch size was changed to 4096. Hence, the batch sizes were considered up to 1024. When the learning rate was changed to other values, mentioned accuracy of the model dropped drastically especially for the learning rate values 0.1, 0.2, and 0.3, that is, accuracy of the model changed from 94.44% to 3.85%; hence, learning rate of 0.001 was considered. For the learning rate of 0.001, batch size of 128, three max-pooling layers, and one average pooling layer, it was found that the result with Relu activation is better. Considering the number of convolutional layers, the experiment has been started with three convolutional layers, and then with four convolutional layers, it was found that the model with four convolutional layers gave a better result. When the number of convolutional layers was changed to 5, the performance of the model decreased. The convolutional layer is followed by batch normalization. Batch normalization was done to speed up training and reduce overfitting.

Following the normalization, max pooling or average pooling with a kernel size of 2×2 is applied, which does a downsampling of the features and sends them to the next layer. Hence, the CNN model with four convolutional layers and two fully connected layers was considered. There was an increase in the accuracy by 2% when the number of fully connected layers was increased from one to two. Further increase in fully connected layers did not increase the accuracy. It was found that the recognition model had good accuracy for the four convolutional layers, ReLu activation function, learning rate =0.001, three max-pooling layers and one layer with average pooling, Batch_size = 128, number of epochs = 300, kernel size 3×3, kernel initializer function = he_normal, dropout = 0.2, and Nadam optimizer. Table 8 shows the accuracy of the CNN model. The possible combinations of max pooling and average pooling were tried.

4.4. Recognition with ResNet18 mix-up regularization

The ResNets (residual neural networks) solve the problem of degradation of DNNs when a large number of convolutional layers are stacked. ResNet18 with mix-up regulation was used for the classification. The architecture is mentioned in Hebbi et al. (2021). The results with dataset 2 were good; to test with Resnet18, dataset 2 was considered. Table 9 shows the results with ResNet18 architecture.

Table 9
Accuracy of ResNet18 model for dataset 2

Upper zone	Middle zone	Lower zone
99.88%	98.92%	97.55%

The proposed models have been cross-validated with the Char74K dataset. It was found the proposed models with the developed dataset gave better results compared to Char74K. Table 10 presents the results of the models on the Char74K dataset.

Table 10
Results with Char74K dataset

SVM with HOG features	CNN	ResNet
98.64	99.45	99.37

Table 11 shows the comparative study of existing work in the field of handwritten Kannada characters for the Kannada language.

Table 11
Comparative study with existing datasets

Authors	Dataset	Character type	# Classes	Samples per class	Model	Accuracy
Prabhu (2019)	Kannada-MNIST	Numerals	10	8300	CNN model	97%
Hebbi et al. (2021)	–	Vowels	13	500	K-means clustering	80%
Aradhya et al. (2010)		Vowels and consonants	50	100	Probabilistic Neural Network	68.89
Parameshwarppa & Dhandra (2015)		Vowels and consonants	48	200	KNN	92.21%
Ramesh et al. (2019)	–	Vowels and consonants	47	500	Capsule Network	98.7%
Rajput & Horakeri (2011)	–	Vowels	13	500	K-NN SVM	91.24% 93.73%
Angadi & Angadi (2015)	–	Vowels and Consonants	49	50	SVM	89.84% and 85.14%
Rao et al. (2020)	Char74K	Vowels, consonants, kagunitas, and numerals	657	25	CNN model	86%
Shobha Rani et al. (2022)	–	Vowels and consonants	49	Varying	Capsule networks	99% for 43 classes
Devaraj et al. (2022)	char74k	Vowels, consonants, kagunitas, and numerals	657+	25	CNN	90%
Chinmayee Bhat (2022)	Kaggle	Vowels, consonants, kagunitas, and numerals	49	25	KNN SVM CNN	72% 88% 96%
Ravikumar & Sampathkumar (2022)	–	Not specified	100+ classes	75	CNN	96.12%
Roopa & Mahantesh (2022)		English alphabet, digits, and Kannada characters	52-English 10 Digits 35 Kannada consonants	Not specified	Graph edit distance	97.01%
Proposed method		Vowels, consonants, modifiers, ottaksharas	83	750	SVM CNN model Resenet18	93.26% 97% 98.78%

*The accuracy of the proposed model mentioned is the average accuracy of each of the zones.

5. Conclusion

This paper presents a detailed description of dataset building for the handwritten Kannada characters which includes vowels, consonants, modifiers, and ottaksharas. K-means clustering, an unsupervised machine learning algorithm, is used to build the dataset. In this approach, 80% of the characters are moved to their bins and the remaining 20% of the characters are manually moved to their respective bins. After the detailed analysis, it was observed that some of the classes were imbalanced. Oversampling and undersampling methods are used to have a balanced number of samples in each class. An oversampling of the model was good compared to the dataset with imbalanced samples and an undersampled dataset. Once the dataset was built, the robustness of the dataset was checked with traditional classifier SVM and HOG features. An accuracy of 99.0%, 88.6%, and 92.2% was obtained for the upper, middle, and lower zones, respectively. Later, the CNN model is fine-tuned with raw input. An accuracy of 100%, 96.15%, and 95.38% was obtained for the upper, middle, and lower zones, respectively. With the ResNet18 model, an accuracy of 99.98%, 97.55%, and 98.92% was obtained for the upper, lower, and middle zones, respectively. It was found that ResNet18 with mix-up regularization also gave good results. The future scope of the research is to recognize the kagunitas and simple and complex words using the dataset built and to add the missing ottaksharas to the dataset.

Acknowledgment

The authors thank the writers who helped in writing the characters for dataset building.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

References

Ahmed, R. M., Rashid, T. A., Fatah, P., Alsadoon, A., & Mirjalili, S. (2021). An extensive dataset of handwritten central Kurdish isolated characters. *Data in Brief*, *39*, 107479.

Alamri, H., Sadri, J., Suen, C. Y., & Nobile, N. (2008). A novel comprehensive database for Arabic off-line handwriting recognition. In *Proceedings of 11th International Conference on Frontiers in Handwriting Recognition, 8*, 664–669.

Ali, H., Ullah, A., Iqbal, T., & Khattak, S. (2020). Pioneer dataset and automatic recognition of Urdu handwritten characters

using a deep autoencoder and convolutional neural network. *SN Applied Sciences*, *2*, 152.

Angadi, S. A., & Angadi, S. H. (2015). Structural features for recognition of handwritten Kannada character based on SVM. *International Journal of Computer Science. Engineering and Information Technology*, *5*(2), 25–32.

Aradhya, M., Niranjana, S. K., & Hemantha Kumar, G. (2010). Probabilistic neural network based approach for handwritten character recognition. *Special Issue of International Journal of Computer and Communication Technology*, *1*(2), 3.

Bartos, G. E., Hoşcan, Y., Kauer, A., & Hajna, É. (2020). A multilingual handwritten character dataset: T-H-E dataset. *Acta Polytechnica Hungarica*, *17*(9), 141–160.

Bellary, S. A. S., & Kusumika, D. (2020). Handwritten kannada alphabets image dataset. Retrieved from: https://sunnybellary.com/project/kannada-dataset/

Biswas, M., Islam, R., Shom, G. K., Shopon, Md., Mohammed, N., Momen, S., & Abedin, A. (2017). BanglaLekha-Isolated: A multi-purpose comprehensive dataset of Handwritten Bangla Isolated characters. *Data in Brief*, *12*, 103–107.

Chinmayee Bhat, H. (2022). Kannada handwritten character recognition using KNN, SVM And CNN. *Journal of Emerging Technologies and Innovative Research*, *9*(8), 454–462.

Das, N., Acharya, K., Sarkar, R., Basu, S., Kundu, M., & Nasipuri, M. (2014). A benchmark image database of isolated Bangla handwritten compound characters. *International Journal on Document Analysis and Recognition*, *17*(4), 413–431.

Dave, D. (2021). Kannada handwritten characters. Retrieved from https://www.kaggle.com/datasets/dhruvildave/kannada-characters

Devaraj, A. Y., Omisha, N., Jain, A., & Shobana, T. S. (2022). Kannada text recognition. *International Journal for Research in Applied Science & Engineering Technology*, *10*(IX), 74–77.

Dongre, V. J., & Mankar, V. H. (2012). Development of comprehensive devanagari numeral and character database for offline handwritten character recognition. Applied Computational Intelligence and Soft Computing, 2012, 1–5.

Ferdous, J., Karmaker, S., Shahariar Azad Rabby, A. K. M. & Hossain, S. A. (2021). MatriVasha: A Multipurpose Comprehensive Database for Bangla Handwritten Compound Characters. In *2020 International Conference on Emerging Technologies in Data Mining and Information Security, 3,* 813–821.

Hebbi, C., Maiya, A., & Mamatha, H. R. (2021). Improving recognition of handwritten Kannada characters using mixup regularization. In *2021 International Advanced Computing Conference, 1528*, 433–447.

Hebbi, C., Metri, O., Bhadrannavar, M., & Mamatha, H. R. (2021). Dataset building for handwritten Kannada vowels using unsupervised and supervised learning methods. In *2020 SIRS 6th International Symposium*, 75–89.

Kannada script (2023). In *Wikipedia.* Retrieved from https://en.wikipedia.org/wiki/Kannada_script

Kavallieratou, E., Liolios, N., Koutsogeorgos, E., Fakotakis, N., & Kokkinakis, G. (2001). The GRUHD database of Greek unconstrained handwriting. In *2001 IEEE 6th International Conference on Document Analysis and Recognition*, 561–565.

Kumar, M., Jindal, M. K., Sharma, R. K., & Jindal, S. R. (2020). Performance evaluation of classifiers for the recognition of offline handwritten Gurmukhi characters and numerals: A study. *Journal of Artificial Intelligence Review*, *53*, 2075–2097.

Lakshmi, B. V., Neelima, Y., Udayani, Y., Satya, L. S., & Parimala, S. J. (2020). A benchmark image database for isolated Telugu handwritten characters. *International Journal of Creative Research Thoughts*, *8*(8), 1154–1161.

Lee, A. W. C., Chung, J., & Lee, M. (2021). GNHK: A dataset for English handwriting in the wild. In *2021 ICDAR 16th International Conference on Document Analysis and Recognition*, 399–412.

Liu, C.-L., Yin, F., Wang, D.-H., & Wang, Q.-F. (2011). CASIA online and offline Chinese handwriting databases. In *2011 IEEE International Conference on Document Analysis and Recognition*, 37–41.

Manjusha, K., Anand Kumar, M., & Soman, K. P. (2019). On developing handwritten character image database for Malayalam language script. *Engineering Science and Technology, an International Journal*, *22*(2), 637–645.

Margaronis, J., Christou, M., Kavallieratou, E., & Tzouramanis, T. (2009). GCDB: A character database system. In *MOCR '09: Proceedings of the International Workshop on Multilingual OCR*, 1–7.

Mohammed, N., Momen, S., Abedin, A., Biswas, M., Islam, R., Shom, G., & Shopon, M. (2017). BanglaLekha-Isolated. In *Mendeley Data.* Retrieved from https://data.mendeley.com/datasets/hf6sf8zrkc/2

Musa, M. E. M. (2011). Arabic handwritten datasets for pattern recognition and machine learning. In *2011 IEEE 5th International Conference on Application of Information and Communication Technologies*, 1–3.

Parameshwarppa, S., & Dhandra, B. V. (2015). A two stage approach for handwritten Kannada character recognition. *International Journal of Engineering Research & Technology*, *3*(19), 1–5.

Parikshith, H., Naga Rajath, S. M., Shwetha, D., Sindhu, C. M., & Ravi, P. (2021). Handwritten character recognition of Kannada language using convolutional neural networks and transfer learning. In *2021 IOP conference series: Materials Science and Engineering*, *1110*, 1–14.

Prabhu, V. U. (2019). Kannada-Mnist: A new handwritten digits dataset for the Kannada language. *arXiv:1908.01242 [cs.CV],* 1–21. Retrieved from https://arxiv.org/abs/1908.01242

Rajput, G. G., & Horakeri, R. (2011). Handwritten Kannada vowel character recognition using crack codes and Fourier descriptors. *In 2011 5th Multi-disciplinary International Workshop On Artificial Intelligence*, 169–180.

Ramesh, G., Manoj Balaji, J., Sharma, G. N., & Champa, H. N. (2019). Recognition of off-line Kannada handwritten characters by deep learning using capsule network. *International Journal of Engineering and Advanced Technology*, *8*(6), 4767–4777.

Rao, A. S., Sandhya, S, Anusha, K., Arpitha, C. N., & Meghana, S. N. (2020). Exploring deep learning techniques for Kannada handwritten character recognition: A boon for digitization. *International Journal of Advanced Science and Technology*, *29*(5), 11078–11093.

Ravikumar, M., & Sampathkumar, S. (2022). Recognition of Kannada handwritten words from answer scripts using machine learning approaches. In *Information and Communication Technology for Competitive Strategies: Applications and Social Interfaces,* 1077–1084.

Roopa, M. J., & Mahantesh, K. (2022). Classification and recognition of bilingual text using graph edit distance based degree of similarity. *Indian Journal of Science and Technology*, *15*(27), 1336–1343

Sagheer, M. W., He, C. L., Nobile, N., Suen, C. Y. (2009). A new large Urdu database for off-line handwriting recognition. In *International Conference on Image Analysis and Processing: Image Analysis and Processiong, 5716*, 538–546.

Shahariar Azad Rabby, A. K. M., Haque, S., Islam, Md. S., Abujar, S., & Hossain, S. A. (2018). Ekush: A multipurpose and multitype comprehensive database for online off-line Bangla handwritten characters. In *2018 International Conference on Recent Trends in Image Processing and Pattern Recognition*, *1037*, 149–158.

Shailesh, A., & Prashnna, G. (2016). Devanagari Handwritten Character Dataset. Retrieved from https://doi.org/10.24432/C5XS53

Shobha Rani, N., Manohar, N., Hariprasad, M., & Pushpa, B. R (2022). Robust recognition technique for handwritten Kannada character recognition using capsule networks. *International Journal of Electrical and Computer Engineering*, *12*(1), 383.

Uddin, I., Ramli, D. A., Khan, A., Bangash, J. I., Fayyaz, N., Khan, A., & Kundi, M. (2021). Benchmark Pashto handwritten character dataset and Pashto Object Character Recognition (OCR) using deep neural network with rule activation function. *Complexity*, *2021*, 1–16.

Velpuru, M. S., Tejasree, G., & Ravi Kumar, M. (2020). Telugu handwritten character dataset. Retrieved from https://dx.doi.org/10.21227/mw6a-d662

How to Cite: Hebbi, C. & Mamatha, H. R. (2023). Comprehensive Dataset Building and Recognition of Isolated Handwritten Kannada Characters Using Machine Learning Models. *Artificial Intelligence and Applications* 1(3), 179–190, https://doi.org/10.47852/bonviewAIA3202624