International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Handwritten Kannada Character Recognition Utilizing CNN, KNN and SVM

Kiran Kumar P N [1], Ranu A[2], Rahul M [2], Sudeep N [2], Pavan T N [2]

[1] Assistant Professor, Dept. of CSE, SJCIT
[2] Student, Dept. of CSE, SJCIT

Abstract

Handwritten character recognition represents a major challenge in the fields of pattern recognition and machine learning, it is more concerned especially in the case of regional languages like Kannada. Kannada, a prominent Dravidian script, poses distinct challenges because of its intricate character formations and diverse handwriting styles. This study introduces a method for identifying handwritten Kannada characters with the application of three machine learning algorithms: K-Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), and Support Vector Machines (SVM). The dataset utilized comprises an extensive array of handwritten Kannada characters sourced from the Kaggle repository. Preprocessing techniques, including image standardization and noise reduction, which are implemented to enhance recognition accuracy. CNN is used for feature extraction as well as classification, while KNN and SVM serve for comparative evaluation. The performance of these models is assessed using metrics such as precision, accuracy, recall, and F1-score. Findings indicate that CNN surpasses KNN and SVM in recognition accuracy, highlights the advantages of deep learning in the classification of handwritten characters. Additionally, a web-based app which is developed using Django to facilitate real-time character recognition. This research significantly contributes to the progress of Optical Character Recognition in Kannada script, promoting advancements in digitization and automation.

Keywords: Kannada Handwritten Character Recognition, Deep Learning, Machine Learning, Optical Character Recognition (OCR), CNN, SVM, KNN, Image Processing, Pattern Recognition.

1. Introduction

Handwritten Character Recognition has become an important area within artificial intelligence and pattern recognition, contributing to the progress of OCR across various languages and scripts. The Kannada script, is prominent Dravidian language, poses distinct challenges owing to its intricate character composition, the existence of compound characters, and the diversity in personal handwriting styles. The increasing demand for digitization and automation in regional languages has necessitated the development of robust machine learning-based recognition systems [1].

Kannada, which is recognized one of the most ancient Dravidian languages, features a script that developed from the Kadamba script and subsequently the Halegannada script. This script comprises 49 fundamental characters, which include both vowels and consonants, also variety of compound characters created through their combination. In contrast to Latin-based scripts, Kannada characters exhibit a more rounded form and feature complex strokes, which can complicate the process of recognition. The intrinsic intricacy of the Kannada script requires advanced recognition models which is capable of adapting to diverse handwriting styles and character formations.

Traditional character recognition methods, including template matching and rule-based strategies, frequently struggle to accommodate the variability present in handwritten text. In order to overcome the limitation, machine learning algorithms, particularly K-Nearest Neighbors (KNN), Convolutional Neural Network (CNN), and Support Vector Machines (SVM), have been used for enhanced classification and accuracy [2]. KNN is a non-parametric approach that performs well with the small datasets; however, it does not scale efficiently when applied to larger datasets. On the other hand, SVM is efficient in handling high-dimensional spaces but may struggle with large-scale datasets when computational complexity increases [3]. Deep learning-based approaches, such as CNN, have demonstrated great performance because of their capability to automatically extract features from input images, making them perfect for tasks involving character recognition [4].

Multiple studies have demonstrated the efficacy of CNN in recognizing handwritten characters in Kannada. Researchers have attained impressive levels of accuracy by employing deep learning frameworks which are trained on extensive datasets. For instance, CNN-based models have demonstrated over 95% accuracy in classifying Kannada characters, significantly outperforming traditional machine learning techniques [5]. However, the combination of multiple classifiers, such as integrating CNN with KNN and SVM, allows for a comparative evaluation of different methodologies, further optimizing recognition performance [6].

The dataset, used in this study is obtained from Kaggle and includes a diverse collection of Kannada handwritten characters. For improving the model's performance, preprocessing methods such as noise reduction, resizing, and normalization are implemented. This approach utilizes Convolutional Neural Network (CNN) for feature extraction and classification, with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) employed as benchmarks for evaluating performance. Additionally, a web-based application using Django has been developed to facilitate real-time recognition of Kannada characters, contributing to advancements in OCR technology for regional languages [7].

This research seeks to overcome deficiencies in recognizing handwritten kannada characters using deep learning techniques. Our research not only improve OCR applications but also facilitate advancements in the digitization of regional scripts, thereby increasing the accessibility and inclusivity of digital content.

2. Related Work

Handwritten Character Recognition is a prominent area of study within OCR. Throughout the years, scholars have devised numerous approaches, spanning from conventional machine learning techniques to models based on deep learning, aimed at improving the precision of recognizing handwritten text. Initial methods for recognizing handwritten kannada characters utilized normal machine learning techniques, such as KNN and SVM. These approaches required significant feature engineering, which included zoning, contour-based descriptors, and moment invariants, to enhance classification accuracy. Samarth et al. [2] investigated KNN-based classification utilizing manually crafted features and achieved satisfactory accuracy. Nonetheless, these conventional methods frequently faced challenges due to differences in handwriting styles and did not possess the necessary robustness for extensive applications.

To overcome these challenges, deep learning models, especially CNN, have gained significant traction in the recognition of Kannada characters. CNNs autonomously extract hierarchical features, thereby removing the necessity for manual feature engineering. Ramesh et al. [1] illustrated the efficacy of CNN in the classification of Kannada characters, attaining greater accuracy than the normal machine learning methods. Their research underscored the benefits of deep learning in managing complex stroke variations and spatial relationships among characters. In the same way, Shobha Rani et al. [5] developed a deep learning framework that surpassed 95% accuracy, emphasizing the significance of convolutional layers in feature extraction and also the classification processes.

The combination of different classifiers which is known as Hybrid models have been used for enhancing the recognition performance. Keerthi Prasad et al. [6] investigated the combination of Principal Component Analysis (PCA) with CNN to refine feature selection and decrease computational demands. Furthermore, Naveena et al. [3] assessed the efficacy of merging CNN with SVM and KNN, showing that these hybrid models can improve accuracy by capitalizing on the advantages of each classifier.

Despite the progress made, challenges persist in the recognition of Kannada characters, including issues related to noisy inputs, variations in stroke thickness, and the scarcity of annotated datasets. Recent studies have concentrated on utilizing transfer learning, data augmentation, and generative models to address these challenges. Future developments in Optical Character Recognition (OCR) for Kannada character may involve the application of transformer-based architectures and graph neural network (GNN) to further improve recognition accuracy.

This research expands on prior studies by performing a comparative analysis of CNN, KNN, and SVM for recognizing handwritten kannada characters.

3. Problem Identification

The recognition of Kannada characters written are handwritten poses considerable difficulties because of the complexity of the script, the diversity of individual handwriting styles, and the existence of visually similar characters. Normal machine learning techniques, such as KNN and SVM, frequently encounter challenges related to high variability within classes and similarities between different classes, resulting in misclassification [1]. Moreover, the scarcity of large, annotated datasets restricts the performance of deep learning models, which necessitate substantial amount of data for effective training [2]. Although CNN have shown enhanced accuracy in character recognition tasks, their high computational demands create obstacles for real-time applications [5]. Additionally, the lack of a standardized methodology for developing Kannada Optical Character Recognition (OCR) systems complicates practical implementation, underscoring the need to investigate a hybrid approach that effectively balances accuracy, computational efficiency, and usability in real-world scenarios [6].

4. Methodology

The proposed system for recognizing handwritten Kannada characters utilizes a hybrid methodology that integrates both machine learning as well as deep learning techniques, specifically KNN, CNN, and SVM. This approach is organized into a systematic pipeline encompassing several stages like dataset collection, preprocessing, feature extraction, model training, classification, performance assessment, and deployment. Additionally, a web-based interface has been created to enable real-time recognition of Kannada characters.

4.1. Dataset Collection

The dataset employed in this research is obtained from Kaggle [8] and comprises thousands of handwritten Kannada character images. Kannada, which belongs to the Dravidian language family, features a complex script that includes 49 fundamental characters which includes 13 Vowels (Swaragalu), 34 Consonants (Vyanjanagalu), 10 numerals, and numerous conjunct consonants. This dataset encompasses handwritten samples from a variety of authors, reflecting differences in stroke thickness, curvature, and individual writing styles.

Figure 1. Workflow of Handwritten Kannada Character Recognition

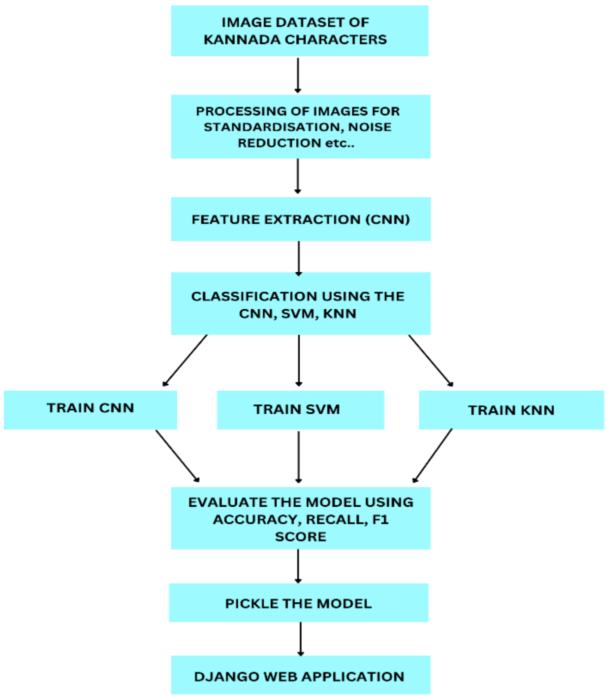


Figure 2. Sample from the dataset

4.2. Data Preprocessing

Preprocessing plays an essential role in the recognition of handwritten characters, as it improves image quality and maintains consistency throughout this dataset. The Kannada script, which is having a curved strokes, varying thickness, and intricate ligatures, necessitates meticulous preprocessing to enhance classification accuracy. The subsequent techniques are employed to enhance the input images prior to training the machine learning models:

- Grayscale Conversion: Images of handwritten characters are typically recorded in RGB (Red, Green, Blue) format, which includes unnecessary color data that is not needed for character recognition. To minimize computational processes and concentrate solely on essential shape and characteristics, all the images are changed into grayscale. This conversion diminishes the channel count from three (RGB) to one, leading to a substantial reduction in memory consumption.
- Image Resizing: Due to the variability in size and proportion of Kannada characters, it is essential to resize all images to a uniform dimension as 32×32 pixels, to ensure consistency throughout the dataset. This process is essential , as machine learning models require images as an input of a fixed size to uphold uniformity and enhance feature extraction. And also, resizing contributes to decreased computation time and lower memory usage.
- Normalization: This method is applied to scale pixel values to a fixed range of [0,1], rather than the standard range of [0,255], to avoid bias towards pixels with greater intensity values. This process enhances the efficiency of CNN, by diminishing the effects of lighting fluctuations and promoting quicker convergence during the training phase.

4.3. Feature Extraction

Feature extraction is essential for character recognition. Various models employ different techniques:

- CNN-Based Feature Extraction: Convolutional Neural Networks (CNNs) autonomously extract hierarchical features, including edges, strokes, and shapes from character images. This model detects spatial patterns, whereas pooling layers help in reducing dimensionality. The introduction of Relu activation adds non-linearity, thereby enhancing the learning of features. Additionally, dropout layers are implemented to mitigate overfitting and improve the generalization of the model.
- Handcrafted features for KNN and SVM: The Histogram of Oriented Gradients (HOG) effectively captures information regarding directional gradients. Zoning features divide the

image into distinct regions to facilitate localized feature extraction. Statistical characteristics, including mean, variance, and skewness, assist in differentiating various writing styles.

4.4. Model Training and Classification

- Convolutional Neural Network (CNN): A deep architecture of CNN comprises several convolutional, pooling, and connected layers. Also, the Adam optimizer is used to train the model, alongside categorical cross-entropy loss. To enhance training stability, batch normalization is employed, while dropout layers are utilized to overcome the risk of overfitting. This network is capable of learning hierarchical representations, which facilitates advanced feature extraction.
- K-Nearest Neighbors (KNN): KNN is a non-parametric classification algorithm assigns a label to an input by examining its nearest labelled neighbors. The algorithm utilizes the Euclidean distance metric to identify the closest match. Although KNN performs well with smaller datasets, it encounters challenges when applied to large-scale data because of its high significant computational demands.
- Support Vector Machine (SVM): This is a technique which converts data into a higher-dimensional space to identify an optimal hyperplane for classification. To address non-linearly separable data, the Radial Basis Function kernel is employed.

Through the comparison of these three classifiers, we aim to ascertain the best method for Kannada character recognition.

4.5. CNN Model Architecture

The proposed CNN architecture is structured with multiple layers aimed at efficient Kannada character categorization and feature extraction. It initiates with three convolutional layers, each accompanied by batch normalization and max-pooling layers, which enhance feature learning and decrease spatial dimensions. The Convolution layers will utilize 32, 64 and 128 number of filters, respectively, facilitating a hierarchical approach to feature extraction. The max-pooling layers serve to downsample the feature maps, thereby minimizing computational complexity while preserving essential information.

Following this, a flattening layer transforms the extracted feature maps into a one-dimensional vector, which is subsequently processed by two fully connected dense layers. The first dense layer contains 256 neurons and is succeeded by batch normalization and dropout layer to overcome the risk of overfitting. The final dense layer, comprising 655 neurons, aligns with the Kannada characters present in the dataset. This model incorporates normalization to enhance learning stability and improve convergence, while dropout regularization contributes to superior generalization during inference.

Figure 3. CNN Model Layers

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	320
batch_normalization (BatchNormalization)	(None, 64, 64, 32)	128
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 32, 32, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 16, 16, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 128)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 256)	2,097,408
batch_normalization_3 (BatchNormalization)	(None, 256)	1,024
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 655)	168,335

4.6. Performance Evaluation

- Accuracy: This metric quantifies the ratio of correctly identified characters.
- Precision: This indicates number of characters predicted correctly among all predicted characters.
- Recall: This evaluates the model's effectiveness in identifying actual characters.
- F1-Score: This metric provides a balance between precision and recall, also thorough evaluation of performance.
- Confusion Matrix: This tool illustrates classification errors and highlights misclassified characters.

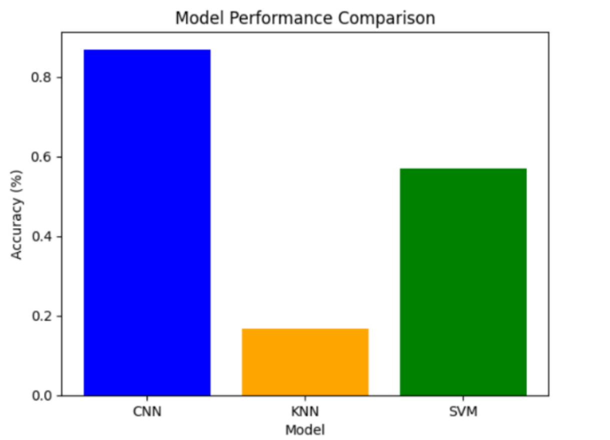
It is anticipated that CNN will surpass KNN and SVM in performance owing to its superior feature extraction abilities; however, utilizing a combination of these classifiers facilitates a comparative analysis.

5. Experimental Results and Comparative Analysis

In order to calculate the performance of the models, the following metrics were used, Accuracy (ACC) which is metric indicates the percentage of characters that were classified correctly. Precision (P) which will represent ratio of relevant instances with total instances retrieved. Recall (R) which measures ratio of similar instances that were successfully identified. F1-Score which is harmonic mean of precision as well as recall, offering a balanced assessment of both metrics.

In order to find the efficacy of our methodology, we have conducted a comparison of the results with previous studies on Kannada handwritten character recognition. Earlier research using normal machine learning techniques, including Random Forest, Decision Trees, and Multi-layer Perceptron (MLP), reported accuracy rates between 85% and 93%. In contrast, our CNN model has achieved an accuracy of 87%.

Figure 4. Model Performance Comparison



Since CNN is best model compared to SVM and KNN, the findings are displayed in the table below:

Table 1. CNN MODEL ANALYSIS

Model	**Accuracy(%)**	**Pression**	**Recall**	**F1-score**
CNN	87	0.90	0.87	0.87

As per the Figure 5. the evaluation of multi-class classification models necessitates a thorough approach to determine their effectiveness. A confusion matrix which serves as an essential tool for illustrating patterns of misclassification among various classes. Essential metrics, including precision, recall, and F1-score, offer a more comprehensive understanding of the model's performance. The model under analysis attained an accuracy rate of 87% and a weighted F1-score of 0.87.

Future improvements is to implement the model with optimization strategies to increase overall performance. Furthermore, techniques for dataset augmentation may be investigated to enhance class representation. Additionally, regularization methods can contribute to the improvement of the model's generalization capabilities.

Figure 5. Confusion Matrix

```
[[1 1 0 ... 0 0 0]
 [1 5 0 ... 0 0 0]
 [0 0 4 ... 0 0 0]
 ...
 [0 0 0 ... 5 0 0]
 [0 0 0 ... 0 6 0]
 [0 0 0 ... 0 0 8]]
```

In the Figure 6. the graph presents a comparison of training accuracy and validation accuracy over 30 epochs for the handwritten kannada character recognition model. At the outset, both training and validation accuracies show a steady increase, reflecting effective learning during the initial phases of training.

Figure 6. Accuracy vs. Epochs Graph

Around the fifth epoch, a significant rise in validation accuracy is observed, followed by some fluctuations, which may indicate potential overfitting or instability in the validation results. As training continues past the tenth epoch, the model reaches a level of stability, achieving an accuracy of approximately 80% for both training and validation datasets. This difference in the validation accuracy during the later epochs will tell us that the model might benefit from fine-tuning techniques, like the implementation of dropout layers or early stopping, to improve its generalization capabilities. Ultimately, the training accuracy is slightly more than the validation accuracy, a typical scenario in the deep learning model which often results from minor overfitting. In short, the model shows encouraging accuracy, further optimization efforts could enhance the consistency of its validation performance.

6. Conclusion

This project effectively employs a CNN for the identification of Kannada characters, utilizing deep learning methodologies to ensure precise classification. The model is subjected to a thorough process of preprocessing, feature extraction, and classification, incorporating CNN, SVM and KNN techniques, with performance assessed through metrics like accuracy, recall, and F1-score. Experimental findings demonstrate that the CNN surpasses conventional machine learning classifiers, achieving superior accuracy and resilience. To strengthen the model further, batch normalization and dropout methods are applied to mitigate overfitting and enhance generalization capabilities. Moreover, the integration of the trained model into a Django-based web application improves its usability, making it suitable for practical applications. Future developments may involve expanding the dataset, investigating more advanced deep learning frameworks, and implementing systems for real-time recognition. Our research contributes for the advancement of effective OCR solutions for low-resource languages, facilitating digital progress in the processing of regional languages.

References

1. Ramesh G., Sharma G.N., Balaji J.M., Champa H.N., "Offline Kannada Handwritten Character Recognition Using Convolutional Neural Networks", 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Bangalore, India, 2019, pp. 1–5. https://doi.org/10.1109/WIECON-ECE48653.2019.9019914
2. Samarth U.K., Joshi S., Chikkamath S., R. N.S., Budihal S., "A Deep Learning Approach for Kannada Handwritten Character Recognition", 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 2024, pp. 1–5.
3. Naveena C., Aradhya V.N.M., "An Impact of Ridgelet Transform in Handwritten Recognition: A Study on Very Large Dataset of Kannada Script", 2011 World Congress on Information and Communication Technologies, Mumbai, India, 2011, pp. 618–621. https://doi.org/10.1109/WICT.2011.6141316
4. Joe K.G., Savit M., Chandrasekaran K., "Offline Character Recognition on Segmented Handwritten Kannada Characters", 2019 Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2019, pp. 1–5. https://doi.org/10.1109/GCAT47503.2019.8978320
5. Rani N.S., Subramani A.C., Kumar P.A., Pushpa B.R., "Deep Learning Network Architecture Based Kannada Handwritten Character Recognition", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 213–220. https://doi.org/10.1109/ICIRCA48905.2020.918316
6. Prasad G.K., Khan I., Chanukotimath N.R., Khan F., "On-line Handwritten Character Recognition System for Kannada Using Principal Component Analysis Approach: For Handheld Devices", 2012 World Congress on Information and Communication Technologies, Trivandrum, India, 2012, pp. 675–678. https://doi.org/10.1109/WICT.2012.6409161
7. Fernandes R., Rodrigues A.P., "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. https://doi.org/10.1109/DISCOVER47552.2019.9008097
8. Dave D., "Kannada Language Characters for OCR Dataset". https://www.kaggle.com/datasets/dhruvildave/kannada-characters