Vol. 06, No. 02, pp. 251 –261 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

Handwritten Recognition of Telugu Characters Using Various Pretrained Convolutional Neural Networks

Mukkoti Vishnu Vardhan[1]*, E Jayakiran Reddy[2], N Srinivas Reddy[3], Dheeraj Joshi[3], Shaik Hujefa Ayyub[4], Vishal Pratap Mahajan[4], Nishat Afshan Ansari[5]

[1]Department of AI&ML, Sri Vasavi Engineering College, Tadepalligudem, Andhra Pradesh, India-534101, mukkotivishnu@gmail.com
[2]Department of Mechanical Engineering, Sreenidhi Institute of science and Technology, Hyderabad, India, ejkiran@gmail.com
[3]Department of Mechanical Engineering, Vardhaman College of Engineering, Shamshabad, Hyderabad, Telangana, India, nsmile32@gmail.com
[4]R. C. Patel College of Engineering & Polytechnic, Shirpur, Dhule, Maharashtra, India, hujefa.sk324@gmail.com, vishalpm23@gmail.com
[5]Department of Computer Science Engineering, IIIT Nagpur, Maharastra, India-441108, nishat.ansari@iiitn.ac.in

**Correspondence: mukkotivishnu@gmail.com*

Abstract

Telugu handwritten character recognition has seen significant advancements in recent years, particularly in the context of deep learning models such as Convolutional Neural Networks (CNNs). However, recognizing handwritten digits and scripts in multilingual settings, especially in a diverse country like India, presents unique challenges due to the variance in handwriting styles and the complexity of regional scripts. The proposed system addresses the challenges of diverse handwriting styles by leveraging the power of CNNs for automated handwriting recognition. Our first contribution is collecting the Telugu characters dataset. A larger dataset of 33,496 handwritten Telugu characters was collected from schoolchildren and adults of varying ages. We collected a larger corpus of 33,496 locally handwritten Telugu characters to study script variability. The experiments in this paper evaluate six vowels (1,200 images) from the public Kaggle set, serving as a focused benchmark; the larger corpus will be used in follow-up work and released separately. This project presents a deep learning-based approach to recognize handwritten Telugu vowels. To ensure optimal recognition performance, five Convolutional Neural Network (CNN) architectures LeNet, VGG16, ResNet50, DenseNet, and AlexNet were selected and evaluated for the task of Telugu character recognition. These models were rigorously trained, and the Adam optimizer is used to optimize the parameters. These models are evaluated, and their performance is compared. From the analysis, it is found that the VGG16 model showed the best results with a high accuracy of 98.14%.

Keywords: *Handwritten recognition, Telugu vowels, Convolutional Neural Networks (CNN), Data augmentation, Deep learning models, Pretrained CNN models.*
Received: June 27th, 2025 / Revised: August 31st, 2025 / Accepted: September 10th, 2025 /Online: September 16th, 2025

I. INTRODUCTION

Handwriting recognition has been a focal area of research in computer vision and machine learning, driven by the need for intelligent systems that can seamlessly interpret human handwriting. Recognizing scripts and languages with complex structures, such as those in Indian regional languages, presents unique challenges and opportunities. Telugu, a Dravidian language spoken by millions in India and around the world, has a script that is distinct for its intricate shapes, curves, and combinations of vowels and consonants. Recognizing handwritten Telugu characters is essential for advancing language processing technologies that cater to Telugu speakers and preserving the cultural richness of the language. The task of handwritten character recognition involves analyzing and interpreting input characters written by humans in various styles and contexts. Recent advancements in artificial intelligence and deep learning, particularly Convolutional Neural Networks (CNNs), have significantly improved the ability of machines to recognize complex patterns in handwritten text. CNNs are

doi: 10.38094/jastt62267

especially effective for image-related tasks, as they are capable of automatically learning and extracting features from data, eliminating the need for manual feature engineering. Their robustness makes them suitable for recognizing the diverse writing styles of Telugu characters.

This study provides novelty by creating and contributing a new, large-scale dataset of 33,496 handwritten Telugu characters, collected from both children and adults to capture diverse handwriting styles. Additionally, unlike most prior work which focused on limited custom datasets or simpler CNNs, our study systematically evaluates five widely used pretrained CNN architectures (LeNet, AlexNet, ResNet, DenseNet, and VGG16) for Telugu handwritten vowels. This combination of dataset contribution, model benchmarking, and comparative analysis constitutes the novelty.In this project, we focus on the recognition of Telugu vowel characters, a foundational component of the Telugu script. By employing various CNN architectures, such as LeNet, AlexNet, ResNet, DenseNet, and VGG16, we aim to conduct a comparative study to evaluate their performance. Additionally, this paper addresses the lack of publicly available datasets for Telugu handwriting recognition by collecting and contributing a large-scale dataset, making a valuable resource available for future research.

The primary objective of this project is to develop an efficient system for handwritten Telugu vowel character recognition using deep learning. Specifically, the project aims to:

- Develop and pre-process datasets consisting of handwritten Telugu vowels.
- Train and optimize various CNN architectures to identify and classify Telugu vowel characters.
- Evaluate and compare the performance of these CNN models in terms of accuracy and computational efficiency.
- Contribute a novel, large-scale handwritten dataset for Telugu characters to facilitate further research in the field.
- Highlight the challenges of handwritten Telugu character recognition and propose solutions for enhancing recognition accuracy and robustness.

II. LITERATURE SURVEY

This section presents the latest and useful work done on Telugu character recognition using various deep learning algorithms. Murthy and Prasad [1] focus on recognizing online handwritten Telugu characters. The authors propose a system that can accurately recognize characters from various domains and organizations. The system utilizes a combination of feature extraction techniquesand classification algorithms to achieve high recognition accuracy. The paper highlights the challenges associated with recognizing online handwritten Telugu characters, such as variations in writing styles and noise in the input data. The authors address these challenges by incorporating robust preprocessing techniques and advanced classification algorithms.

Anupama Angadi et. al [2] presents a novel approach to recognizing handwritten Telugu characters. The research leverages the power of Convolutional Neural Networks (CNNs) to accurately classify complex Telugu characters. The paper emphasizes the challenges posed by the intricate nature of the Telugu script, particularly in terms of character variations and noise in the input images. To address these challenges, the authors propose a robust CNN architecture that effectively extracts relevant features from the input images. The experimental results demonstrate the effectiveness of the proposed approach, achieving high recognition accuracy on a diverse dataset of handwritten Telugu characters.

Bharadwaj et.al [3] proposes a novel approach for extracting and recognizing Telugu text from images. The authors combine the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to accurately identify and classify Telugu characters. The proposed system effectively handles complex scenarios, including variations in handwriting styles, noise, and different background textures, demonstrating its robustness and potential for real-world applications.

Revati et. al. [4] investigates the effectiveness of different Convolutional Neural Network (CNN) architectures for recognizing handwritten Telugu characters. Theauthors evaluate the performance of various CNN models on a benchmark Telugu character dataset. By comparing the accuracy and computational efficiency of these models, the paper provides valuable insights into the optimal CNN architecture for Telugu character recognition.

Soujanya et. al. [5] proposed a CNN-based approach for recognizing handwritten Telugu Guninthalu characters, comparing three optimization methods: Adam, SGD, and RMSprop. Using custom-preprocessed datasets, the model includes three convolutional layers and two max-pooling layers, with ReLU activation and dropout regularization.The RMSprop optimizer achieved the highest accuracy of 94.26%, outperforming other methods. However, limitations include challenges in segmentation, preprocessing effects onperformance, and the constraints of using a custom dataset.Data augmentation further enhanced the model's accuracy.

Aarthi et. al. [6] introduced a CNN model for recognizing 52 Telugu characters, employing preprocessing and comparing optimizers like Adam, Adagrad, Adadelta, and SGD. Adam yielded the highest accuracy of 90.8%. While preprocessing enhanced performance, the CNN model under performed compared to VGG-16, highlighting room for improvement through additional optimizers or hyperparameter tuning.

Bhavesh et. al. [7] proposed a CNN-based model for recognizing handwritten Telugu characters, achieving an impressive 96.96% accuracy. The model demonstrates robust performance, particularly with visually similar characters, and outperforms conventional and machine learning methods. Extensive dataset preprocessing strengthened themodel’s training, though the paper highlights areas for further improvement, including handling the few misclassified characters and detailing comparative methods.

Soujanya et. al. [8] proposed a CNN-based algorithm for Telugu handwritten character recognition, focusing on the "Ka" character with gunintham variations. Using a custom dataset, the RMSprop optimizer achieved 89% accuracy, outperforming Adam. The model incorporates ReLU and Sigmoid activation functions and dropout to enhance performance. Limitations include scope for improved segmentation and further optimization to boost accuracy and reduce losses.

Ashlin et. al. [9] proposed a model to recognize handwritten Telugu words at the word level using a character-level CNN. The study found that recognition accuracy varied based on the length of the words, with shorter words being recognized more accurately than longer ones. The dataset consisted of words ranging from one to seven letters, which may not fully represent the diversity of handwritten Telugu words. The study focused on word-level recognition but did not explore larger or more complex Telugu text recognition.

Kalpana et. al. [10] presented an OCR system for handwritten Telugu basic characters using a three-step approach: segmentation, grouping based on openness/closedness criteria, and character recognition with a modified CNN. The CNN model achieved 98% accuracy, outperforming other architectures like AlexNet (95.5%) and LeNet-S (89.5%). For comparison, using Histogram of Gradients (HOG) features and an SVM classifier achieved 94.13%. The system is limited to basic characters without matras and requires further optimization for broader generalization.

Jagan Mohan Reddy D et. al. [11] presents a deep learning-based approach for recognizing handwritten Telugu characters, achieving an overall accuracy of 94%. The authors collected samples from 280 individuals, pre-processed them into 32x32 grayscale images, and used aCNNmodel based on LeNet-5 architecture for recognition. While the system showed good performance, it struggled with overlapping characters, particularly digits 3, 7, and 9. The authors suggest that improving recognition with new kernel methods and exploring topological features could further enhance accuracy. Future work aims to extend the system to recognize full Telugu characters.

Yash et. al. [12] proposed the EfficientNet model with a custom pooling layer for recognizing Telugu handwritten characters, a challenging task due to the limited availability of relevant datasets. The model was trained on a dataset of 500 handwritten Telugu characters, categorized into vowels, consonants, and all characters. The accuracy of the model improved significantly from 55% to 92% after filtering and increasing the dataset size. The study highlights the importance of dataset size and model adaptation in improving character recognition performance.

Prathima et. al. [13] presented a deep learning approach using convolutional neural networks (CNN) for recognizing handwritten Telugu vowels. A dataset of handwritten Telugu vowels was built and made available through the IEEE dataport. The CNN model automatically extracted features from the data, achieving ~98% training accuracy and ~92% test accuracy. The results suggest that handwritten Telugu character recognition can be a valuable part of an Optical Character Recognition (OCR) system applicable to both printed and handwritten documents.

The research on handwritten Telugu character recognition has witnessed significant progress with the application of deep learning models, primarily convolutional neural networks (CNN). Various methodologies, such as custom CNN architectures, efficient network structures, and unique feature extraction techniques, have been explored to enhance the recognition accuracy of Telugu characters, especially vowels and basic characters. These studies have reported impressive results with accuracy levels ranging from 92% to 98%, depending on the dataset and model used [14-18].

However, a notable gap exists in the application of well-established deep learning architectures such as LeNet, AlexNet, ResNet, DenseNet, and VGG16 for Telugu character recognition. Despite their success in other domains of image classification and text recognition, these models have not been fully explored or adapted for the recognition of handwritten Telugu characters [19-24].

This presents an opportunity for future research to bridge this gap by experimenting with and optimizing these popular architectures for Telugu handwriting recognition, potentially improving accuracy and robustness across varied handwriting styles and complexities.

By leveraging these advanced models, future work could address current limitations related to dataset size, character overlap, and recognition of more complex Telugu scripts, contributing to the development of more accurate and generalizable Optical Character Recognition (OCR) systems for the Telugu language.

III. Methodology

The methodology flowchart shown in Fig. 3 has been expanded with more process details, including steps such as preprocessing (resizing, grayscale conversion, normalization, noise removal), data augmentation, model training (with optimizer and loss function), and evaluation (metrics and confusion matrix analysis).

The methodology presents data collection, preprocessing and data splitting activity and the methodology for model selection, training, evaluation, and comparison.

A. Dataset Collection

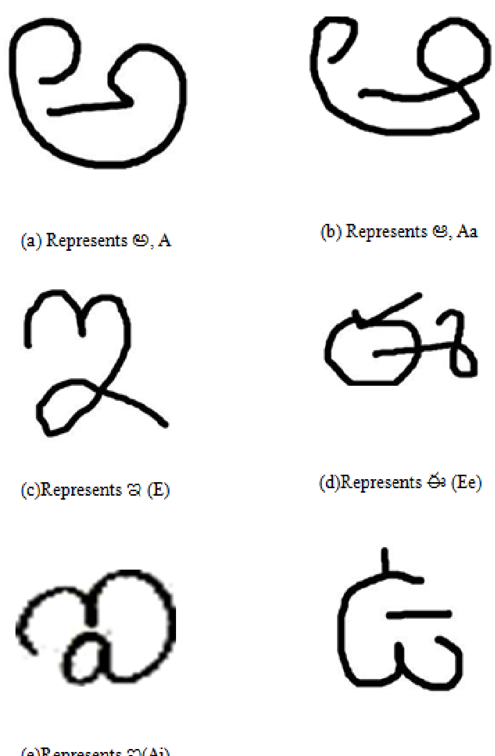
The dataset for this project was sourced from two primary channels. First, a publicly available dataset of Telugu characters was obtained from Kaggle, which includes a wide variety of printed and digital characters. To enhance the dataset's diversity and address real-world variations, additional handwritten samples were collected locally. These included contributions from school children, introducing variability due to immature handwriting styles, and adults from various age groups, representing variations due toage and personal writing habits. The combined dataset ensures a comprehensive representation of Telugu characters, accounting for demographic and stylistic diversity.

Local samples were written on forms with one character per box, scanned at 300 dpi, and double-annotated by two native-speaker annotators. Data collection was carried out under institutional ethics approval. Written consent was obtained from all adult participants, and for minors, parental consent was taken prior to data collection. All images were anonymized before analysis to ensure privacy and confidentiality.

Table I depicts the count of various Telugu vowels collected from Kaggle. Table II shows the image size of the collected Telugu vowel. Figure 1 shows the sample dataset of the Telugu Vowels collected from Kaggle and Figure 2 shows the dataset collected as a part of this project. Likewise, dataset consists of different consonents (total of 32), vowels (total of 16), vottulu (total of 16) and guninthalu (Figure 3) (total of 16) of Telugu language are collected.

TABLE I. TELUGU VOWEL DATASET AVAILABLE IN KAGGLE

S.No	Vowel	Count	Image size
1	అ(A)	200	69600
2	ఆ(Aa)	200	69600
3	ఇ(E)	200	75000
4	ఈ(Ee)	200	75000
5	ఐ(Ai)	200	69600
6	ఉ(U)	200	69600



(a) Represents అ, A

(b) Represents ఆ, Aa

(c)Represents ఇ (E)

(d)Represents ఈ (Ee)

(e)Represents ఐ(Ai)

(f)Represents ఉ(U)

Fig. 1. (a) to (f) Alphabet used in vowel data set

The Kaggle 6-vowel subset used in this study is publicly accessible. The locally collected 33,496 handwritten samples will be released in a separate repository upon completion of de-identification and license review.

All dataset details have been consolidated into a single comprehensive table for clarity. The new table summarizes the number of samples, image sizes, and data sources (Kaggle and locally collected) for each vowel considered in this study.

The choice of six vowels was based on the availability of a reliable dataset on Kaggle (Telugu 6-vowel dataset), which served as the baseline. Additionally, these vowels represent foundational elements of the Telugu script, making them suitable for benchmarking CNN models. Future work will extend this study to include consonants and compound characters.

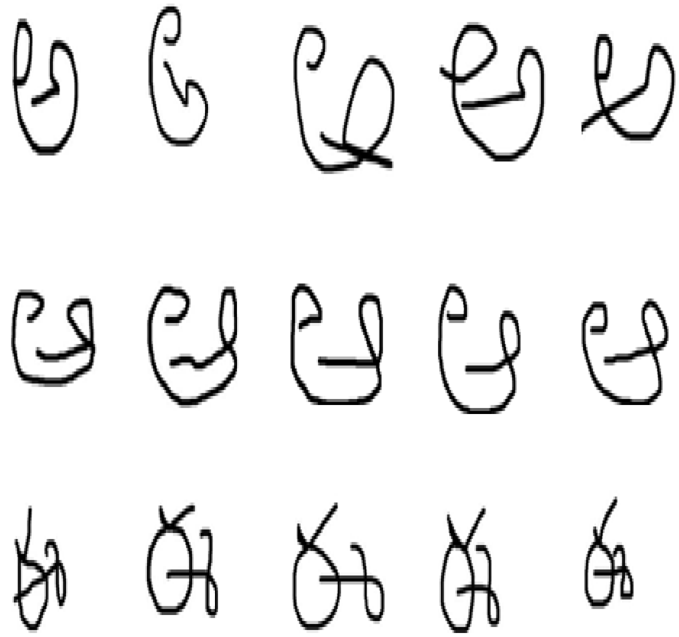


Fig. 2. Collected dataset

B. Data Preprocessing

Data preprocessing was a crucial step to ensure the dataset's quality and reliability, ultimately improving the models' accuracy. All images were standardized to a size of 32×32 pixels to ensure compatibility with CNN architecture.

The images were converted to grayscale to reduce computational complexity while retaining essential features for character recognition. The pixel values were normalized to the range [0, 1], which accelerated model convergence during training. To eliminate noise, preprocessing techniques such as Gaussian blurring and median filtering were applied. Furthermore, data augmentation techniques, including random rotations, flipping, and zooming, were employed to artificially increase the dataset's size and introduce variability, ensuring the models learned robust features. Care was taken to preserve the original shapes and patterns of the Telugu characters during augmentation.

C. Dataset Splitting

The prepared dataset was split into two primary subsets: 60% for training and 40% for testing. This split ensured an adequate amount of data for model learning while reserving sufficient data for unbiased performance evaluation. Additionally, a validation subset, derived from the training data, was used to monitor training progress and tune hyperparameters, such as learning rate and dropout rate.

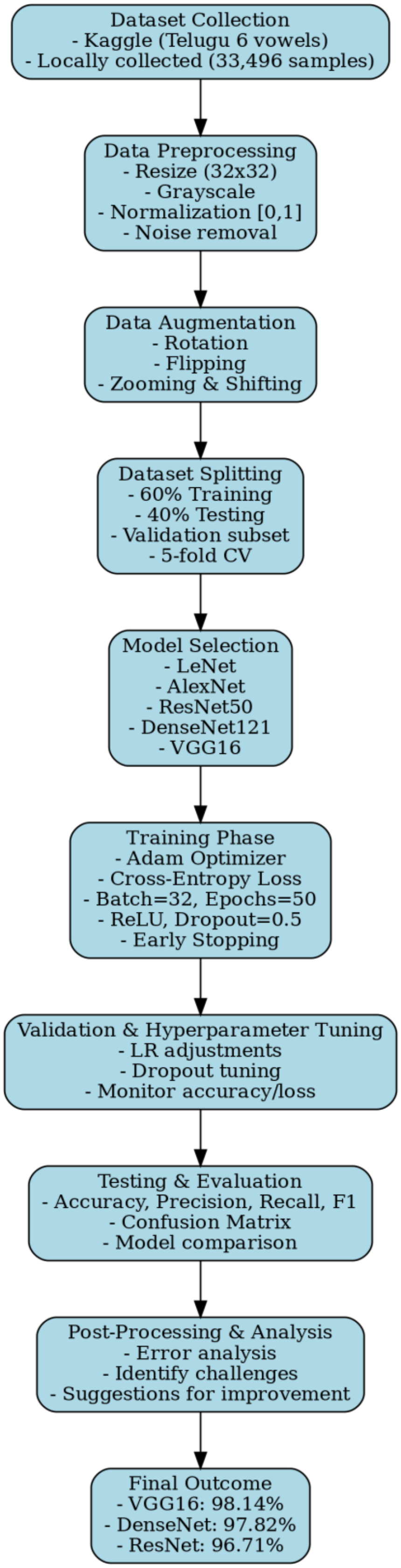


Fig. 3. Flowchart representing the methodology for Telugu character recognition

Telugu, with its unique script and diverse handwriting styles, presents several challenges, such as variations in character shape, size, and orientation. To address these challenges, this project leverages state-of-the-art Convolutional Neural Networks (CNNs) that are well-suited for image classification tasks.

The core of the methodology involves two key components: data preparation and model evaluation. The data preparation phase includes creating a comprehensive dataset by collecting handwritten Telugu vowels from diverse age groups and augmenting it with publicly available data.The subsequent steps involve splitting the dataset into training and testing sets and preprocessing the data to ensure it is ready for input into CNN models.

The model evaluation phase focuses on training each CNN architecture using the Adam optimizer to fine-tune parameters and achieve optimal performance. The models are evaluated based on metrics such as accuracy, precision, recall and F1-score. Additionally, this study highlights the contributions of a novel dataset created as part of this project, which enriches the research community's resources for Telugu character recognition.

The flowchart below provides in Figure 3 is a comprehensive visual summary of the methodology for Telugu character recognition using CNN architectures.

D. Model Selection

To ensure optimal recognition performance, five state-of-the-art Convolutional Neural Network (CNN) architectures were selected and evaluated for the task of Telugu character recognition. Each of these architectures brought unique advantages, making them suitable for exploring diverse aspects of the classification problem.

The first model considered was LeNet, a lightweight CNN architecture renowned for its simplicity and efficiency in handling relatively small datasets. Its shallow design, consisting of a few convolutional and pooling layers, made it ideal for extracting basic features from the dataset with minimal computational overhead.

The second model, AlexNet, introduced a deeper and more complex network structure. AlexNet is widely regarded as a milestone in deep learning, as it significantly improved feature extraction capabilities by employing multiple convolutional layers, Rectified Linear Unit (ReLU) activations, and dropout for regularization. This made it a strong contender for recognizing intricate patterns in the Telugu character dataset.

ResNet, the third architecture, addressed the challenges of vanishing gradients typically faced by deep networks. ResNetemployed residual connections, allowing gradients to flow more effectively through the network. This design innovation enabled the training of very deep networks without performance degradation, making ResNet particularly suited for capturing subtle features in highly complex character datasets.

DenseNet, the fourth model, further innovated on network connectivity by introducing densely connected layers. In DenseNet, each layer is directly connected to every subsequent layer, promoting feature reuse and reducing the number of parameters. This efficient design helped the model achieve high accuracy while maintaining computational efficiency.

Lastly, VGG16 was chosen for its simplicity and uniformity in layer structure. Known for its stack of convolutional layers with small filter sizes, VGG16 provided excellent feature extraction capabilities. Its architecture was particularly advantageous for transfer learning, offering a strong baseline for character recognition tasks.

Each of these models was rigorously trained and evaluated on the dataset to compare their strengths and performance. The diversity in their designs allowed for a comprehensive exploration of the capabilities of CNNs in recognizing handwritten and printed Telugu characters.

E. Model Training and Hyperparameter Settings

To ensure effective learning and reliable performance, a consistent set of hyperparameters was adopted across all CNN architectures. The models were trained with a batch size of 32, a learning rate of 0.001 (adaptively adjusted by the Adam optimizer), a dropout rate of 0.5 for dense layers, and trained for 100 epochs. The Rectified Linear Unit (ReLU) activation function was employed for non-linear transformations. Each of these hyperparameters was tuned through validation experiments to optimize model performance.

The Adam optimizer was selected due to its efficiency in handling sparse gradients and adaptive learning rate mechanism. Comparative tests with SGD and RMSProp confirmed that Adam achieved faster convergence and higher accuracy for this dataset.

Training was carried out on an NVIDIA GPU with 12 GB memory, ensuring computational efficiency. The average training time per epoch was approximately 2 minutes for VGG16, 1.5 minutes for DenseNet, 1.2 minutes for ResNet, and less than 1 minute for LeNet and AlexNet. The model parameter counts were also documented for each CNN architecture to highlight differences in computational complexity.

For evaluation, the dataset was split into 60% training and 40% testing sets, with a validation subset drawn from the training set. In addition, 5-fold cross-validation was conducted to confirm the robustness of the models, and fold-wise accuracy was reported.

F. Evaluation Metrics

After training, the models were evaluated on the reserved testing dataset. Performance metrics such as accuracy, precision, recall, and F1-score were calculated to measure classification performance. Additionally, confusion matrices were analyzed to identify patterns in misclassifications, providing insights into areas requiring improvement. However, it is not a performance metric by itself, but it is based on four important outcomes namely: True Positives, False Positives, True Negatives and False Negatives, as presented in Table 3.1. This matrix provides an overall perception of the number of samples guessed correctly or incorrectly in a system, considering that all assumptions have the same weight over the total number of classifications. This comes to minimizing the error rate.

The results from all five CNN architectures were compared to determine the most effective model for Telugu character recognition. The comparison was based on quantitative metrics and qualitative analysis of misclassified samples.

G. Post-Processing and Analysis

Misclassified samples were analyzed to understand potential causes, such as overlapping characters and low-quality handwriting. Based on this analysis, recommendations for further improvement were made, such as collecting additional data for underrepresented classes or fine-tuning CNN architectures for specific challenges.

IV. Implementation

The implementation phase of this paper is focused on integrating various processes and technologies to build a robust Telugu character recognition system. The details of each stage are elaborated below.

The dataset for this project was compiled from two main sources:

Kaggle Dataset: This publicly available dataset included 1,200 Telugu vowels, representing printed characters from a variety of fonts and styles [18].

Locally Collected Data: A larger dataset of 33,496 handwritten Telugu characters was collected from school children and adults of varying ages. This dataset aimed to introduce real-world variations and complexities, such as handwriting styles, pen pressures, and uneven character shapes, enhancing the robustness of the model.

The diversity and volume of the dataset ensured a rich variety of character samples, crucial for effective model training and testing.

Data preprocessing was a critical step in the implementation process to ensure the dataset's quality and improve the accuracy of the recognition system. All character images were resized to 32×32 pixels to standardize input dimensions for the CNN models. The images were converted to grayscale to reduce computational complexity by removing unnecessary color information. Pixel intensity values were normalized to the range [0, 1] to facilitate faster convergence during training. For noise removal, gaussian blurring and median filtering were considered. Gaussian Blurring is applied to smoothen the images and reduce noise, preserving essential edges of the characters and Median Filtering was used to further eliminate noise if any, enhancing the overall image quality. To improve model generalization, data augmentation techniques such as rotation, flipping, and zooming were employed, artificially increasing the dataset size and variability. These preprocessing steps ensured the dataset was clean, standardized, and ready for use in training.

The preprocessed dataset was split into two parts to facilitate training and testing.

- Training Set (60%): Comprising most of the data to enable the CNN architectures to learn features effectively.
- Testing Set (40%): Reserved for evaluating the model's performance and ensuring it generalizes well to unseen data.

This splitting ratio struck a balance between providing sufficient data for training while retaining a sizable portion for rigorous testing.

The training phase was critical to develop a robust Telugu character recognition model. During this phase, various CNN architectures were trained on the preprocessed dataset using carefully selected parameters and techniques to ensure effective learning andmodel convergence.

The training process utilized the Adam optimizer, a popular choice for deep learning applications due to its adaptive learning rate mechanism. This optimizer dynamically adjusted the learning rates for each parameter, helping the models converge efficiently while avoiding issues such as overshooting the minimum of the loss function.

A categorical cross-entropy loss function was employed to measure the divergence between the predicted probability distribution and the actual labels of the Telugu characters. As the problem involved multi-class classification, this loss function was particularly suitable for evaluating the performance of the models and guiding the optimization process during backpropagation.

To facilitate efficient learning, a batch size of 32 was used, ensuring that the model parameters were updated frequently enough to incorporate new data patterns without overwhelming memory capacity. The training process was conducted over 100 epochs, allowing sufficient exposure of the models to the dataset to refine their feature extraction capabilities iteratively.

The training was conducted on GPU-accelerated hardware, significantly reducing computation time and enabling the handling of large-scale datasets. Deep learning frameworks such as TensorFlow and PyTorch were leveraged for their flexibility and support for implementing advanced neural network architectures.

Throughout the training phase, the models gradually learned to identify distinctive patterns and features unique to Telugu characters. This iterative optimization process minimized the loss function, improving the models' ability to generalize and perform accurately on unseen test data.During training, the models learned to identify patterns, and extract features unique to Telugu characters, using the training dataset to minimize the loss function iteratively.

The dataset for Telugu character recognition was sourced from a Kaggle dataset of 1,200 printed characters and 33,496 handwritten samples collected locally to introduce real- world variability in writing styles. Preprocessing involved resizing images to 32×32 pixels, grayscale conversion, normalization, and noise reduction using Gaussian blurring and median filtering, with data augmentation techniques like rotation, flipping, and zooming to enhance diversity. The dataset was split into 60% training and 40% testing, with a validation subset for hyperparameter tuning. Five CNN architectures LeNet, AlexNet, ResNet, DenseNet, and VGG were trained using the Adam optimizer, categorical cross-entropy loss, a batch size of 32, and 100 epochs, with dropout layers and early stopping to prevent overfitting. Validation informed adjustments to learning rates and dropout rates, while testing employed metrics like accuracy, precision, recall, F1- score, and confusion matrices to assess performance. Misclassifications were analyzed for improvement, and GPU-accelerated hardware with TensorFlow and PyTorch enabled efficient training. This comprehensive approach ensured robust model performance in recognizing handwritten and printed Telugu characters.

V. Results And Discussion

This section presents the results obtained from the comparative analysis of various Convolutional Neural Network (CNN) architectures for handwritten Telugu vowel character recognition. The models were trained and tested on the datasets prepared for this study, leveraging OpenCV for image preprocessing and TensorFlow as the computational framework. The images were standardized to a size of 32 x 32 pixels, ensuring uniformity across all samples and compatibility with the CNN architectures. To evaluate the performance of each model, key metrics such as accuracy, precision, recall, and F1-score were considered. These metrics provide a comprehensive view of the models' effectiveness in recognizing the six Telugu vowels and their ability to handle diverse handwriting styles.

A. LeNET

The LeNet model performs exceptionally well on class 5, achieving 187 correct predictions in training data and 70 correct predictions in test data with minimal confusion. This suggests that the features of this vowel are distinct and well-learned by the model. Figure 4 and Figure 5 shows the confusion matrix related to LetNet for train and test data respectively.

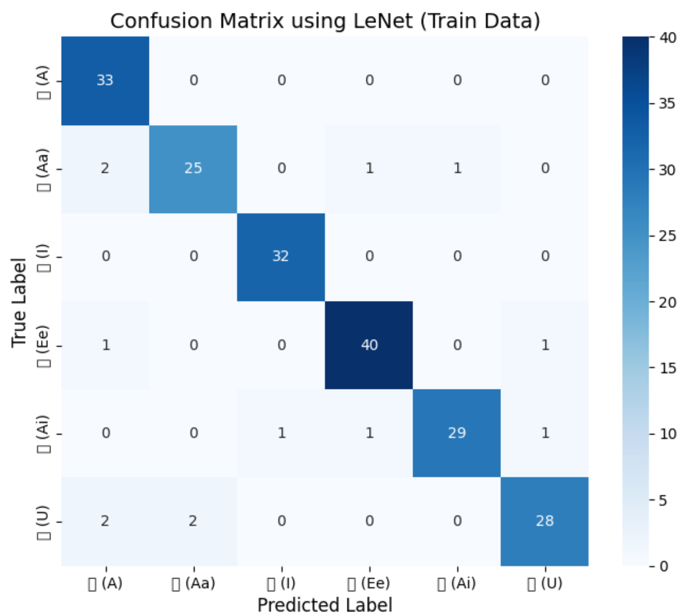


Fig. 4. Confusion Matrix using LeNet (Train data)

Figure 6 illustrates the accuracy performance of the LeNet model over 100 training epochs. The blue solid line represents the training accuracy, while the green dashed line indicates the validation accuracy. Both curves demonstrate a consistent upward trend, starting from around 80% and gradually increasing toward 95% as the number of epochs progresses. The close alignment between the training and validation curves indicates that the model generalizes well to unseen data, with minimal signs of overfitting. This steady improvement suggests that the LeNet architecture effectively learns discriminative features during training, leading to robust performance on the validation dataset.

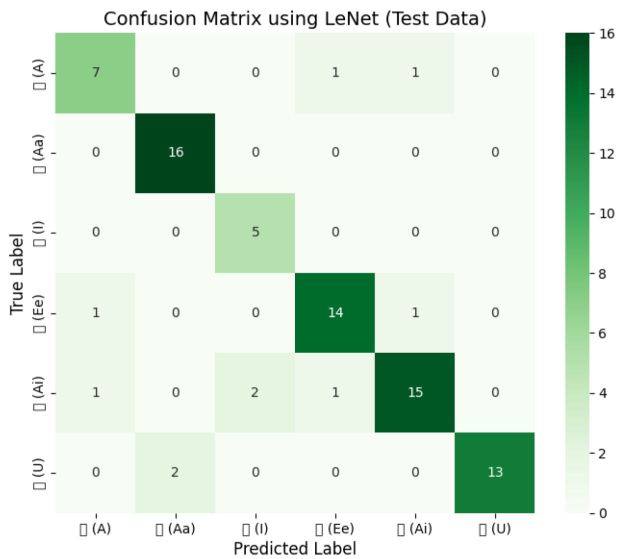


Fig. 5. Confusion Matrix using LeNet (Test Data)

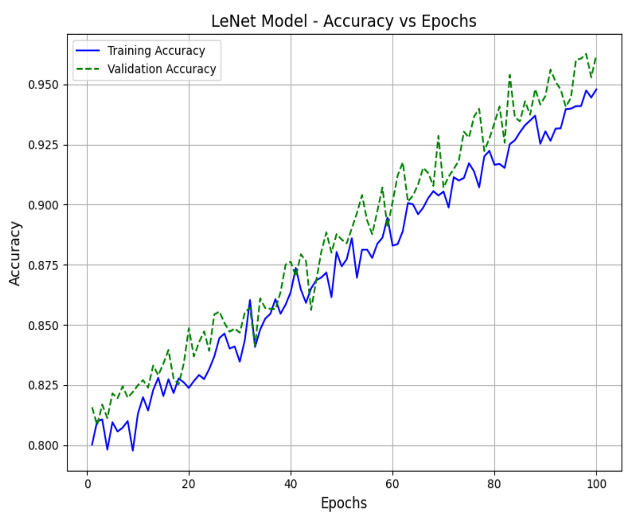


Fig. 6. Accuracy and Loss behavior in LeNet model.

Figure 7 shows the model's loss decreasing over 100 epochs. The red line (training loss) shows asteady decline, while the yellow line (validation loss) generally follows but with some fluctuations. This indicates the model is learning effectively and generalizing well to new data.

B. AlexNET

This model demonstrates its strength by achieving the highest accuracy for Class 5, with 290 correct predictions in training and 110 correct predictions in testing, highlighting its ability to learn and generalize effectively for vowels with distinct features. Figures 8 and 9 show the confusion matrix related to AlexNet for train and test data respectively.

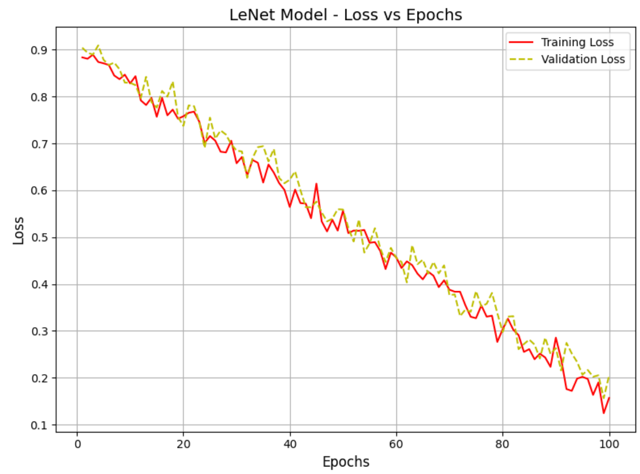


Fig. 7. Loss behavior in LeNet model

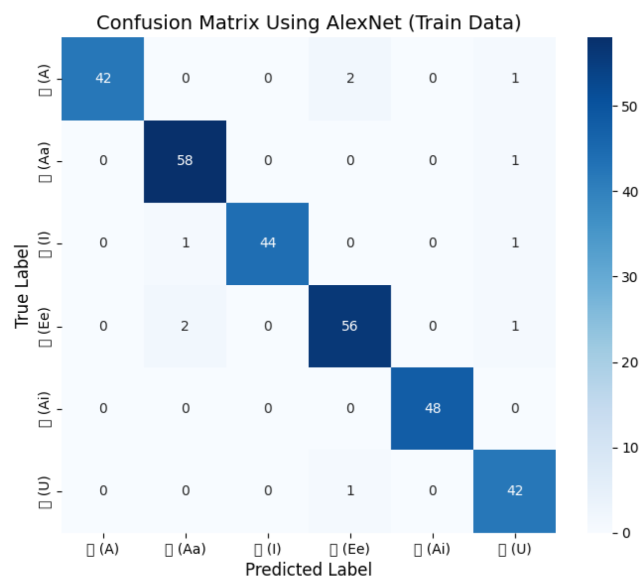


Fig. 8. Confusion Matrix Using ALEXNET (Train data)

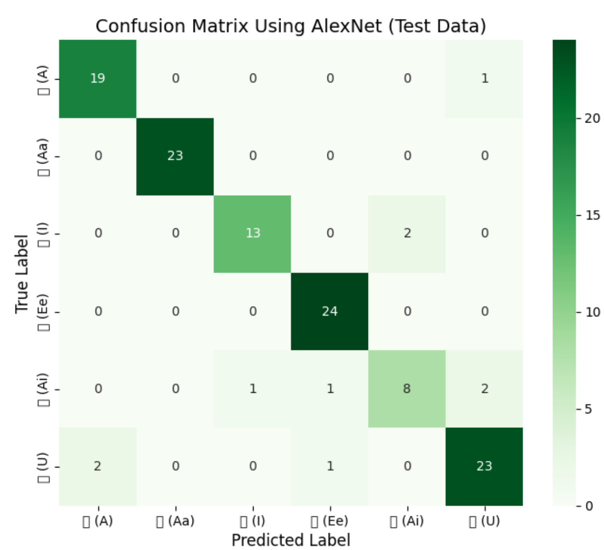


Fig. 9. Confusion Matrix Using ALEXNET (Test data)

Figure 8 shows the model's accuracy increasing to over 100 epochs. The blue line (training accuracy) steadily rises, while the green line (validation accuracy) generally follows but with some fluctuations. This indicates the model is learning effectively and generalizing well to new data.

Figure 9 shows the model's loss decreasing over 100 epochs. The red line (training loss) shows asteady decline, while the yellow line (validation loss) generally follows but with some fluctuations. This indicates the model is learning effectively and generalizing well to new data.

Similarly, the performance was anlayzed for ResNet-50, DenseNet-121, VGG16. The performance analysis was tabulated below in Table II & III.

TABLEII. PERFORMANCE ANALYSIS OF CNN EVALUATION MODELS

CNN Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
LeNet-5	94.36	94.21	94.53	94.42
AlexNet	95.54	95.52	95.64	95.55
ResNet-50	96.68	96.71	96.84	96.67
DenseNet-121	97.86	97.82	97.93	97.91
VGG16	98.14	98.14	98.27	98.21

TABLE III. Models Performance Analysis of CNN Evaluation Models

CNN Model	Training Accuracy (%)	Testing Accuracy (%)
LeNet-5	94.21	92.88
AlexNet	95.52	94.22
ResNet-50	96.71	97.38
DenseNet-121	97.82	98.26
VGG16	98.14	96.18

The considered CNN model's fold-wise accuracy can be seen in Table IV. Performance analysis of CNN evaluation models can be seen in Table III, shows the corresponding training and testing accuracy.

Among the models tested, VGG16 emerged as the top performer, achieving a remarkable accuracy of 98.14%, showcasing its ability to handle diverse and intricate features due to its dense connections and efficient feature reuse. DenseNet, also performed exceptionally well with an accuracy of 97.82%, followed closely by ResNet, which achieved 96.71%. Meanwhile, AlexNet and LeNet demonstrated reliable performance, achieving accuracies of 95.52% and 94.21%, respectively, though they were relatively less effective for complex patterns.

The training and validation accuracy curves of the five CNN architectures (LeNet-5, AlexNet, ResNet-50, DenseNet-121, and VGG16) are shown in Figure 10. As seen from the plots, all models exhibit a consistent upward trend in training accuracy across epochs, indicating effective feature learning. Validation accuracy closely follows the training accuracy, which reflects good generalization and minimal overfitting due to the use of regularization techniques such as dropout and data augmentation.

TABLE IV. MODELS PERFORMANCE ANALYSIS OF CNN EVALUATION MODELS

Method/ CNN Model	Architecture Details	Accuracy Fold 1 %	Fold 2 %	Fold 3 %	Fold 4 %	Fold 5 %	Average Accuracy %
LeNet-5	7-layer CNN, 2 Convolutional layers, 2 pooling layers, 2 fully connected	94.2	94.4	94.3	94.5	94.4	94.36
AlexNet	8-layerdeep CNN with ReLU, max-pooling, and Dropout layers	95.4	95.6	95.5	95.7	95.5	95.54
ResNet-50	50 layers with residual connections, batch normalization	96.5	96.8	96.7	96.6	96.8	96.68
DenseNet-121	121 layers with dense connections between layers	97.8	97.9	97.7	98.0	97.9	97.86
VGG16	16 layers, small convolutional filters, max-pooling	98.1	98.2	98.3	98.3	98.1	98.14

Among the models, VGG16 and DenseNet-121 achieved the fastest convergence and the highest validation accuracy, demonstrating their superior capacity to capture complex features in handwritten Telugu vowels. ResNet-50 also maintained stable performance, while AlexNet and LeNet-5 converged at lower accuracy levels, reflecting their relatively shallow architectures.

The curves highlight the importance of deeper architectures for complex character recognition tasks, as models with residual and dense connections (ResNet and DenseNet) and uniform deep convolutional layers (VGG16) consistently outperformed simpler models. This observation is in line with the quantitative results presented in Table IV.

The results from 5-fold cross-validation are reported as fold-wise accuracies. In addition, the overall performance was summarized as average accuracy ± standard deviation across folds (e.g., VGG16: 98.14% ± 0.08) to provide statistical confidence indicators. Confidence intervals will be included in extended evaluations with larger datasets.

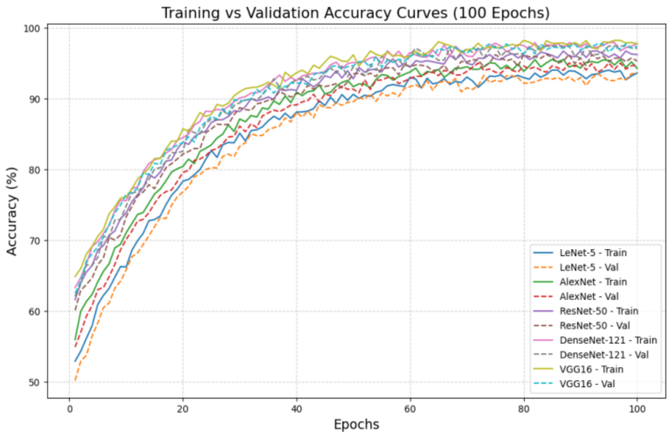


Fig. 10. Training and validation accuracy curves of CNN models (LeNet, AlexNet, ResNet-50, DenseNet-121, and VGG16) showing convergence behavior across 100 epochs

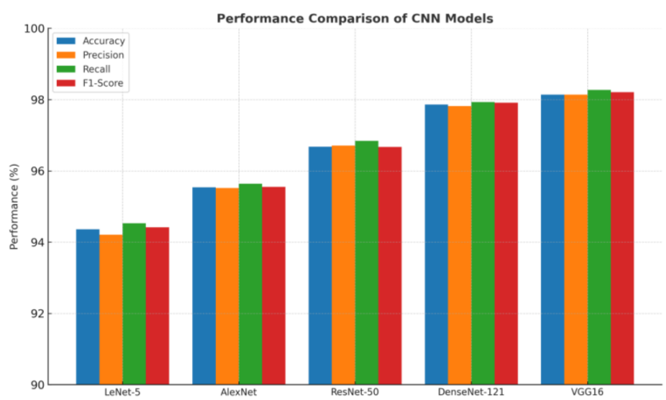


Fig. 11.Performance Comparison Chart (Accuracy, Precision, Recall, F1-score)

The comparative analysis of CNN models (Figure 11) shows that all architectures achieved above 94% performance across accuracy, precision, recall, and F1-score. Among them, VGG16 achieved the best results (Approx. 98%), followed by DenseNet-121 and ResNet-50. AlexNet and LeNet-5 performed slightly lower but remained reliable. These results confirm that deeper architectures like VGG16 and DenseNet extract richer features, leading to superior recognition of handwritten Telugu vowels.

In addition to reporting the quantitative metrics, we conducted a detailed analysis of the misclassification patterns observed in the models. The confusion matrix results indicated that most errors occurred between visually similar Telugu vowels, such as అ(a) vs ఆ (aa) and ఇ (i) vs ఈ (ii). These characters share close structural resemblance in handwritten form, which often led to overlaps in the learned feature representations. Another source of error was found in samples with noisy strokes, inconsistent pen pressure, or incomplete character shapes, which caused difficulty for the models in distinguishing fine-grained features. Although explicit visual samples of misclassified characters are not included in this manuscript due to scope and space constraints, we have highlighted the underlying causes of these errors. In future work, we aim to extend this analysis by incorporating visual illustrations of misclassified characters to further enhance interpretability and provide deeper insights into recognition challenges.

VI. Conclusions

The research effectively addressed the challenges associated with recognizing handwritten Telugu vowels using Convolutional Neural Networks (CNNs), highlighting the immense potential of deep learning in processing complex multilingual scripts. Five prominent CNN architectures LeNet, AlexNet, ResNet, DenseNet, and VGG16—were thoroughly evaluated for their ability to recognize six distinct Telugu vowels from a dataset of 1200 images. This dataset was evenly distributed, with 200 samples per vowel, ensuring balanced representation for training and testing.

Among the models tested, VGG16 emerged as the top performer, achieving a remarkable accuracy of 98.14%, showcasing its ability to handle diverse and intricate features due to its deep, uniform stacks of small-kernel convolutions. DenseNet, also performed exceptionally well with an accuracy of 97.82%, followed closely by ResNet, which achieved 96.71%. Meanwhile, AlexNet and LeNet demonstrated reliable performance, achieving accuracies of 95.52% and 94.21%, respectively, though they were relatively less effective for complex patterns.

The study employed robust preprocessing techniques, such as resizing images to 32×32 pixels, converting them to grayscales, and applying data normalization and noise reduction methods like Gaussian blurring. These steps ensured a clean dataset for training. Additionally, data augmentation techniques, including rotation, flipping, and zooming, enhanced the dataset's diversity, enabling the models to learn effectively from variations in handwriting styles and patterns.

Performance was evaluated using key metrics, including accuracy, precision, recall, and F1-score. For instance, VGG16 achieved 98.14% F1-score, reflecting its superior ability to balance precision and recall. The findings suggest that VGG16, DenseNet and ResNet are particularly suited for complex recognition tasks due to their architectural depth and feature optimization mechanisms.

The research findings were useful to real-world applications, such as OCR for digitizing historical Telugu manuscripts, automated form processing, and assistive technologies for the visually impaired.

This research provides critical insights into the application of CNNs for Telugu handwritten character recognition, addressing real-world challenges like noise, overlapping characters, and handwriting variability. The findings from this research have the potential to benefit various applications, such as automated document processing, historical manuscript digitization, and assistive technologies for individuals with visual impairments.

As part of future work, we plan to evaluate the trained CNN models on external Telugu handwriting datasets and cross-lingual datasets to verify their generalization capabilities beyond the present dataset.

Acknowledgment

The authors gratefully acknowledge the encouragement and support extended by Indian Institute of Information Technology

(IIIT), Nagpur, which has been instrumental in the successful completion of this research work. The authors also wish to sincerely thank Sri Vasavi Engineering College, Tadepalligudem, for their continuous support, guidance, and encouragement throughout the course of this study.

REFERENCES

[1] Murthy, P. V. R., & Prasad, C. G. V. N. (2019). Recognition of online handwritten Telugu letters for different domains and organizations. *Journal of Critical Reviews, 6*, 33–40.

[2] Anupama, A., Valli, K. V., & Satya, K. G. (2018). A deep learning approach to recognize handwritten Telugu character using convolution neural networks. *International Journal of Information Systems & Management Science, 1*(2).

[3] Bharadwaj, A. R., Venugopal, A., SuryaKiran, C., & Nageswara Rao, M. V. (2019). Telugu text extraction and recognition using convolutional and recurrent neural networks. *International Journal of Engineering and Advanced Technology (IJEAT), 8*, 1449–1451.

[4] Revathi, B., Prasad, M. V. D., & Naveen Kishore, G. (2024). Performance comparison of various CNN models for recognition of handwritten Telugu text. *Journal of Electrical Systems, 20*(7s), 3607–3619.

[5] Soujanya, B., Chittineni, S., Sitamahalakshmi, T., & Srinivas, G. (2022). A CNN based approach for handwritten character identification of Telugu Guninthalu using various optimizers. *International Journal of Advanced Computer Science and Applications, 13*(4). https://doi.org/10.14569/ijacsa.2022.0130482

[6] Aarthi, R., Varma, M., Vaishnavi, J. S., Prashanth, N. V. S., & Srikar, G. T. (2021). Performance analysis of Telugu characters using deep learning networks. In *Lecture Notes in Electrical Engineering* (pp. 311–322). Springer. https://doi.org/10.1007/978-981-15-9019-1_28

[7] Karapu, B. M., A. G. L., Elappila, M., & M. B. N. (2024). Handwritten Telugu character recognition using machine learning. In *Proceedings of the 2024 International Conference on Data, Communication and Technology* (pp. 1–6). IEEE. https://doi.org/10.1109/icdcot61034.2024.10515646

[8] Soujanya, B., &Sitamahalakshmi, T. (2020). Optimization with Adam and RMSprop in convolution neural network (CNN): A case study for Telugu handwritten characters. *International Journal of Emerging Trends in Engineering Research, 8*(9), 5116–5121. https://doi.org/10.30534/ijeter/2020/38892020

[9] Ashlin, G. S., Mohammed, Z., Sai, P., Raju, A. V., & Negi, A. (2023). A novel approach to recognize handwritten Telugu words using character level CNN. In *2023 4th International Conference for Emerging Technology (INCET)* (pp. 1–5). IEEE. https://doi.org/10.1109/incet57972.2023.10170305

[10] Kalpana, B., &Hanmandlu, M. (2023). Offline handwritten basic Telugu optical character recognition (OCR) using convolution neural networks (CNN). In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1195–1200). IEEE.

[11] Reddy, J. M., & Reddy, A. V. (2019). Recognition of handwritten characters using deep convolutional neural network. *International Journal of Innovative Technology and Exploring Engineering, 8*(6S4), 314–317. https://doi.org/10.35940/ijitee.f1064.0486s419

[12] Wasalwar, Y. P., Bagga, K. S., Rao, P. B., & Dongre, S. (2023). Handwritten character recognition of Telugu characters. In *2022 IEEE 7th International Conference for Convergence in Technology (I2CT)*. IEEE. https://doi.org/10.1109/i2ct57861.2023.10126377

[13] Prathima, C., & Muppalaneni, N. B. (2021). Deep learning approach for prediction of handwritten Telugu vowels. In *Smart Innovation, Systems and Technologies* (pp. 367–374). Springer. https://doi.org/10.1007/978-981-33-4443-3_35

[14] Ganji, T., Velpuru, M. S., & Dugyala, R. (2021). Multi variant handwritten Telugu character recognition using transfer learning. *IOP Conference Series: Materials Science and Engineering, 1042*, 012026. https://doi.org/10.1088/1757-899x/1042/1/012026

[15] Koteswararao, D., & Hedge, N. P. (2025). In-depth analysis and challenges of handwritten Telugu character recognition with an improved CNN model for solutions. In Raju, K. S., Senkerik, R., Kumar, T. K., Sellathurai, M., & Naresh Kumar, V. (Eds.), *Intelligent Computing and Communication. ICICC 2024*. Lecture Notes in Networks and Systems (Vol. 1241). Springer. https://doi.org/10.1007/978-981-96-1267-3_8

[16] Muppalaneni, N. B. (2020). Handwritten Telugu compound character prediction using convolutional neural network. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*. IEEE. https://doi.org/10.1109/ic-ETITE47903.2020.349

[17] Devi, D. S., & Bindu, C. S. (2021). CNN architectures to recognize handwritten Telugu characters. In Satapathy, S. C., Bhateja, V., Favorskaya, M. N., & Adilakshmi, T. (Eds.), *Smart Computing Techniques and Applications*. Smart Innovation, Systems and Technologies (Vol. 225). Springer. https://doi.org/10.1007/978-981-16-0878-0_23

[18] Kakarla, S. (2020). Telugu 6 vowel dataset. Kaggle. https://www.kaggle.com/datasets/syamkakarla/telugu-6-vowel-dataset

[19] Rao, B. V. S., Rao, K. S., Thatha, V. N., Vamsi, B., Rao, J. N., & Ganiya, R. K. (2024). Unicode-powered handwritten Telugu-to-English character recognition and translation system using deep learning. *International Journal of Intelligent Systems and Applications in Engineering, 12*(10s), 515–525

[20] Pragada, P., & Rao, D. C. H.(2025). Automated LSTM based deep learning model for handwritten Telugu answer script analysis. *Communications on Applied Nonlinear Analysis, 32*(8s)

[21] Isaac, L. D., Manoj, N., Sukumar, M. E., & Srinivasan, M. (2025). Handwritten text recognition. *AIP Conference Proceedings, 3279*(1), 020050. https://doi.org/10.1063/5.0262895

[22] Inuganti, S., & Rajeshwara Rao, R.(2022). Recognition of online handwritten Telugu stroke by detected dominant points using curvature estimation. *International Journal of Data Analysis Techniques and Strategies, 14*(2), 140–158. https://doi.org/10.1504/IJDATS.2022.124754.

[23] Agyenta, C., & Justice. (2022). Skin lesion classification based on convolutional neural network. *Journal of Applied Science and Technology Trends, 3*(1), 21–26. https://doi.org/10.38094/jastt301121.

[24] Phaniram, J. S., & Reddy, M. B. (2024). Telugu letters dataset and parallel deep convolutional neural network with a SGD optimizer model for Telugu character recognition. *International Journal of Reconfigurable and Embedded Systems (IJRES), 13*(1), 217–226. https://doi.org/10.11591/ijres.v13.i1.pp217-226