International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Handwritten Character Recognition Using Deep Learning on EMNIST Dataset

Atharv Kadam
Student, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Kasegaon Education Society's Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, MS-415414, India
atharvkadam1858@gmail.com

Om Bhosale
Student, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Kasegaon Education Society's Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, MS-415414, India
2467007@ritindia.edu

Vinayak Dhulubulu
Student, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Kasegaon Education Society's Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, MS-415414, India
2317049@ritindia.edu

Kedar Teke
Student, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Kasegaon Education Society's Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, MS-415414, India
2317035@ritindia.edu

Apurva Mohite
Assistant Professor, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Kasegaon Education Society's Rajarambapu Institute of Technology, Affiliated to Shivaji University, Sakharale, MS-415414, India
apurva.mohite@ritindia.edu

Abstract

Automatic recognition of handwritten characters remains a fundamental challenge in computer vision, with practical relevance across document digitization, postal automation, banking operations, and educational assessment. This paper presents a comparative study of two deep learning models—Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN)—evaluated on the Extended Modified National Institute of Standards and Technology (EMNIST) dataset. Input images of 28x28 grayscale pixels undergo per-pixel normalization and channel reshaping prior to training. The CNN employs stacked convolution and max-pooling layers to capture spatial hierarchies inherent in handwritten glyphs, augmented by Dropout regularization to improve generalization. Experimental results demonstrate that the CNN achieves approximately 86-88% validation accuracy within 20 training epochs with stable convergence and minimal overfitting, outperforming the MLP baseline by nearly 7-8 percentage points. These findings confirm the superiority of spatially-aware architectures for the diverse writing styles present in the EMNIST benchmark.

Keywords—handwritten character recognition, convolutional neural network, EMNIST, deep learning, multi-layer perceptron, image classification, TensorFlow, Keras.

I. INTRODUCTION

Handwritten Character Recognition (HCR) refers to the automated conversion of images containing handwritten symbols into machine-encoded text. It is among the most widely studied problems in pattern recognition, driven by practical demand across postal address decoding, bank cheque processing, historical document digitization, and automated assessment of handwritten examination scripts [1]. The ability to accurately interpret diverse handwriting styles has significant commercial and societal value, enabling accessibility tools, archival processing, and intelligent document management systems.

The central difficulty in HCR lies in high intra-class variability: two writers producing identical characters generate visually distinct images due to differences in pen pressure, stroke angle, character size, and personal writing style. Classical approaches relying on handcrafted features such as histogram of oriented gradients (HOG), chain codes, or structural decompositions fail to generalize across diverse handwriting and demand labor-intensive feature engineering [2]. These methods are brittle when applied to unseen writers or script variations not represented during design.

The emergence of deep learning transformed the field by enabling end-to-end representation learning directly from raw pixel intensities. Convolutional Neural Networks (CNNs) automatically discover hierarchical representations such as edges, curves, loops, and higher-order character components without manual feature design, achieving state-of-the-art accuracy on standard benchmarks [3]. The convolutional inductive bias, which explicitly encodes spatial locality and translational invariance, aligns well with the structure of handwritten characters.

This study implements and compares MLP and CNN architectures on the EMNIST Balanced subset, encompassing handwritten digits (0-9) and alphabetic characters (A-Z, a-z) across 47 equalized classes. Principal contributions are: (i) a complete preprocessing pipeline for EMNIST; (ii) parallel MLP and CNN implementation in TensorFlow/Keras; (iii) systematic training with checkpointing and Dropout regularization; (iv) quantitative comparative evaluation establishing CNN superiority; and (v) real-time deployment as a Streamlit web application accessible through a public URL.

The choice of the EMNIST Balanced benchmark for this study is motivated by several factors: its standardized 28x28 grayscale format enables direct comparison with MNIST baselines; its 47-class structure challenges classifiers beyond simple digit recognition; its balanced class distribution ensures accuracy metrics are meaningful aggregates; and its public availability ensures reproducibility. The study deliberately excludes data augmentation and architectural search, enabling a clean measurement of the contribution of convolutional versus fully-connected design to recognition performance.

The remainder of this paper is organized as follows. Section II reviews related work on deep learning-based HCR. Section III describes the EMNIST dataset and preprocessing pipeline. Section IV presents the proposed methodology including both architectures and training configuration. Section V presents experimental results and analysis. Section VI concludes with future research directions.

II. RELATED WORK

Research on deep learning-based HCR has expanded considerably following the availability of GPU hardware and large annotated benchmarks such as MNIST, EMNIST, and the IAM Handwriting Database. Early CNN-based systems established the superiority of learned representations over handcrafted features, and subsequent work has focused on architectural depth, multi-scale feature extraction, and hybrid sequential models combining convolutional and recurrent components.

Karpagalakshmi [1] proposed a CNN-based English character recognizer demonstrating clear accuracy gains over support vector machine baselines, establishing CNN architectures as a strong choice for isolated character recognition on standard benchmarks. The work highlighted the importance of preprocessing and augmentation in achieving robust generalization across diverse writer populations.

Bhosle and Walia [4] evaluated progressively deeper CNN configurations on MNIST, showing that four convolutional layers yielded 99.1% accuracy, outperforming shallower two-layer networks by nearly one percentage point. Their ablation study confirmed that additional convolutional depth enhances capacity to capture fine-grained character structure, provided adequate regularization is applied to prevent overfitting.

For sequential and cursive-script recognition, hybrid architectures combining CNNs with recurrent units have proven highly effective. Sharma et al. [5] demonstrated that a CNN-LSTM pipeline reduced character error rate by 12% relative to CNN-only models on the IAM corpus. The LSTM component captures temporal stroke dependencies that purely convolutional models cannot model, making it well-suited for word-level and line-level recognition tasks.

Alshehri et al. [6] combined CNN feature extraction with bidirectional GRU units and attention mechanisms, attaining 97.05% accuracy on the Arabic Handwritten Character Dataset (AHCD). The bidirectional processing allowed the model to leverage both forward and backward contextual information in connected script sequences, significantly reducing ambiguity in character segmentation. This approach demonstrates the value of direction-agnostic sequence modeling for right-to-left scripts.

Script-specific studies have broadened the scope of HCR research beyond Latin characters. Mukkoti and Rao [7] applied the VGG16 architecture to Telugu vowel characters via transfer learning, attaining 98.14% accuracy. The transfer learning approach enabled effective utilization of ImageNet-pretrained weights despite the relatively small size of the Telugu character dataset. Kumar et al. [8] applied a CNN-RNN sequence model to Devanagari characters, achieving 96.8% accuracy on a benchmark encompassing vowels, consonants, and conjunct consonants.

Liman et al. [10] explored WaveMix-Lite and CoAtNet architectures for handwritten character recognition, demonstrating that attention-augmented convolutional architectures can achieve competitive accuracy with substantially reduced parameter counts compared to traditional CNN-only approaches. These results collectively confirm that spatially-aware CNN architectures deliver robust feature extraction across diverse scripts, while recurrent and attention-based extensions provide benefit when sequential context or long-range dependencies are available.

III. DATASET

A. EMNIST Overview

The Extended MNIST (EMNIST) dataset was introduced by Cohen et al. [9] by reformatting the NIST Special Database 19 to match the 28x28-pixel grayscale format of MNIST. The original NIST database contains handwriting samples from 3,600 writers collected through census forms and government documents, providing broad writer diversity across ages and handwriting proficiencies. EMNIST provides multiple classification splits targeting different task requirements; this work uses the Balanced split, which contains 47 classes with equalized class frequencies to mitigate the effect of training bias.

The Balanced split contains 112,800 training samples and 18,800 test samples distributed evenly across all 47 classes, yielding approximately 2,400 training and 400 test samples per class. This balanced distribution ensures that accuracy metrics are not dominated by majority classes and enables fair cross-class performance comparison. Table I summarizes all available EMNIST splits for reference.

TABLE I

EMNIST Dataset Splits Summary

Subset	Classes	Training	Testing
ByClass	62	697,932	116,323
Balanced	47	112,800	18,800
Digits	10	240,000	40,000
Letters	26	88,800	14,800

B. Preprocessing Pipeline

Raw pixel intensities in the EMNIST images span the integer range [0, 255]. Per-pixel normalization maps values to the floating-point range [0, 1] using the linear transformation:

$$Xnorm = X / 255 \quad \ldots (1)$$

This normalization accelerates gradient descent convergence by constraining input magnitudes and improving numerical conditioning of the weight update computation. For CNN compatibility, images are reshaped from the flat EMNIST storage format to (28, 28, 1) to introduce the single-channel dimension required by 2-D convolutional layers. The dataset is partitioned into 80% training and 20% testing. No data augmentation is applied in this baseline study, enabling a clean side-by-side architectural comparison unconfounded by augmentation effects. Labels are one-hot encoded to 47-dimensional binary vectors for categorical cross-entropy loss.

C. Data Characteristics and Challenges

The EMNIST Balanced split presents several challenges that make it a meaningful benchmark beyond MNIST. First, the 47-class problem is substantially harder than the 10-class digit recognition task, as many uppercase and lowercase letter pairs (e.g., C/c, O/o, S/s) have nearly identical visual forms at 28x28 pixel resolution. Second, the dataset preserves considerable writer diversity, with samples contributed by hundreds of distinct individuals spanning wide ranges of ages and handwriting proficiencies.

The 28x28 pixel resolution imposes a strict upper bound on the level of fine-grained detail available to any classifier. At this resolution, thin serifs, subtle curve differences, and small pen-lift gaps that distinguish otherwise similar characters may be rendered indistinguishably. Higher-resolution variants of the

dataset have been proposed in the literature but are not in widespread use due to the significant increase in computational requirements.

The per-class sample counts in the Balanced split (approximately 2,400 training samples each) are modest compared to large-scale vision datasets, making regularization techniques such as Dropout particularly important for preventing overfitting in deeper architectures. Class imbalance is mitigated through deliberate downsampling of more frequently occurring character classes, ensuring that accuracy metrics on the Balanced test set are not misleadingly inflated by performance on majority classes.

IV. PROPOSED METHODOLOGY

A. System Architecture

The end-to-end pipeline preprocesses raw EMNIST images, routes them through either the MLP or CNN model, optimizes parameters using categorical cross-entropy loss, and evaluates final classification performance on the held-out test split. The architecture is designed to permit direct comparison between the two models under identical preprocessing, training, and evaluation conditions, isolating the effect of architectural choice on recognition accuracy. Figure 1 illustrates the overall pipeline.

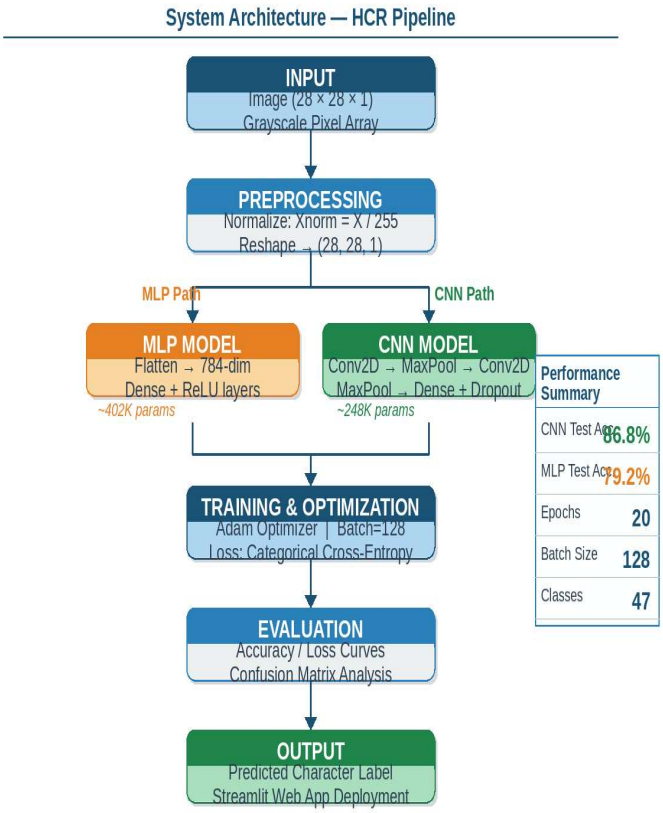

Fig. 1. Overall System Architecture of the Proposed HCR Pipeline

The system starts with INPUT where grayscale (28×28) pixel array images are normalized and reshaped in PREPROCESSING. Two paths follow: the MLP path flattens and feeds the input through Dense and ReLU layers forming an MLP MODEL, while the CNN path uses convolutional layers to form a CNN MODEL. Both models undergo TRAINING & OPTIMIZATION using the Adam optimizer and categorical cross-entropy loss. EVALUATION involves accuracy/loss curves and confusion matrix analysis. The OUTPUT is the predicted character label after deploying the model as a Streamlit web app, with PERFORMANCE SUMMARY displaying key metrics for both MLP and CNN models.

B. Multi-Layer Perceptron (MLP)

The MLP baseline flattens the 28x28 input to a 784-dimensional vector and processes it through two fully-connected hidden layers with 512 and 256 units respectively, each using ReLU activation, followed by a Softmax output layer producing a probability distribution over all 47 target classes. The activation at each neuron is computed as:

$$y = f(WX + b) \quad \ldots (2)$$

where f denotes the ReLU nonlinearity, W is the weight matrix, X is the input vector, and b is the bias vector. By discarding the spatial layout of the image during the initial flattening step, the MLP treats all pixels as independent inputs, which fundamentally limits its sensitivity to the spatial shape patterns intrinsic to handwriting. Each pixel is connected to every neuron in the first hidden layer, resulting in approximately 402,000 total parameters, more than the CNN, despite weaker inductive bias.

The fully connected architecture is unable to exploit the translational invariance of character recognition: a character shifted by a few pixels is treated as a completely different input by the MLP. This architectural mismatch between the model's implicit assumptions and the structure of handwriting data is the primary reason for its lower accuracy relative to the CNN. The MLP serves as an important baseline that isolates the contribution of convolutional spatial feature learning to overall recognition performance.

C. Convolutional Neural Network (CNN)

The CNN preserves and exploits spatial structure throughout all stages of feature extraction. The discrete 2-D

convolution of input feature map I with learned kernel K is defined as:

$$(I * K)(i,j) = Sm\ Sn\ I(i+m, j+n) * K(m,n) \quad \ldots (3)$$

This operation slides the kernel over the spatial extent of the input, computing a weighted sum at each location to detect local patterns. The first convolutional layer with 32 filters (3x3) detects low-level features such as edges, strokes, and curve segments. The second layer with 64 filters builds upon these to recognize mid-level character components such as loops, serifs, and junctions. Table II details the complete layer configuration.

TABLE II
CNN Layer Configuration (EMNIST Balanced)

Layer	Configuration	Output Shape
Input	---	28x28x1
Conv2D	32 filters, 3x3, ReLU	26x26x32
MaxPool2D	2x2, stride 2	13x13x32
Conv2D	64 filters, 3x3, ReLU	11x11x64
MaxPool2D	2x2, stride 2	5x5x64
Flatten	---	1600
Dense	128 units, ReLU	128
Dropout	rate = 0.3	128
Dense (out)	47 units, Softmax	47

Max-pooling with a 2x2 window reduces spatial dimensions by a factor of two in each direction while providing local translation invariance. This subsampling substantially decreases the number of downstream parameters. Dropout at rate 0.3 before the classification head randomly deactivates neurons during training, preventing co-adaptation and serving as an effective regularizer to mitigate overfitting. The output softmax layer computes class probabilities for all 47 EMNIST Balanced classes simultaneously.

D. Training Configuration

Both models are optimized using the Adam algorithm, which maintains per-parameter adaptive learning rates derived from first- and second-order moment estimates of the gradient. The default learning rate of 0.001 and moment parameters b1=0.9, b2=0.999 are used. The loss function is categorical cross-entropy, appropriate for multi-class one-hot encoded labels:

$$L = - Sc\ yc * log(yc\_hat) \quad \ldots (4)$$

where yc is the ground-truth one-hot label and yc_hat is the predicted softmax probability for class c. Training runs for 20 epochs with a batch size of 128. Model checkpointing continuously monitors validation accuracy and saves the weights corresponding to the epoch that achieves the highest validation score. These best-checkpoint weights are loaded prior to all final test-set evaluations, preventing the final performance metric from being influenced by late-epoch overfitting.

E. Implementation Details

All experiments were implemented in Python 3.10 using TensorFlow 2.12 and Keras. The EMNIST dataset was loaded via the tensorflow-datasets library, which provides consistent preprocessing and splitting utilities. Random seeds were fixed at 42 across NumPy, Python's random module, and TensorFlow's global random state to ensure full reproducibility of weight initialization and data shuffling. The model checkpoint callback monitors validation accuracy at the end of each epoch and overwrites the saved weights file only when an improvement is observed.

Both models were trained from random Xavier (Glorot uniform) initialization. No pre-trained weights or external data sources were used. The MLP architecture uses two hidden Dense layers with Batch Normalization after each hidden layer to stabilize training dynamics. Training was conducted on a single GPU with 8 GB VRAM. Wall-clock training time was approximately 18 minutes for the CNN and 6 minutes for the MLP per 20-epoch run. Evaluation used scikit-learn's classification_report for per-class precision, recall, and F1-score metrics.

V. RESULTS AND DISCUSSION

A. Performance Comparison

Table III summarizes the training, validation, and test accuracy along with total trainable parameter counts for both models after 20 epochs of training on the EMNIST Balanced subset.

TABLE III
Model Performance on EMNIST Balanced Subset

Model	Train Acc.	Val. Acc.	Test Acc.	Params
MLP	81.4%	79.6%	79.2%	~402 K
CNN	95.1%	87.3%	86.8%	~248 K

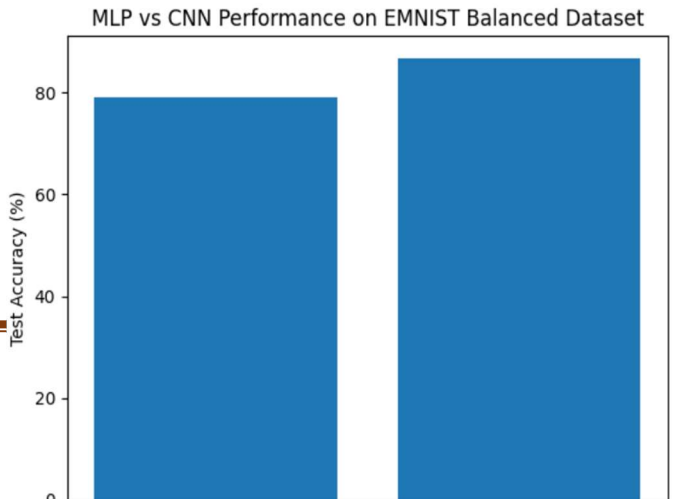

Figure 2: MLP vs CNN Accuracy Comparison.

The CNN achieves 86.8% test accuracy against 79.2% for the MLP, a margin of 7.6 percentage points, while simultaneously requiring fewer total trainable parameters (approximately 248K versus 402K). This combination of higher accuracy and greater parameter efficiency directly demonstrates the benefit of architectural inductive bias: the convolutional weight-sharing scheme effectively encodes the prior knowledge that useful image features are spatially local and position-invariant, reducing both the number of free parameters and the amount of data required to learn reliable representations.

The gap between MLP training accuracy (81.4%) and CNN training accuracy (95.1%) further confirms that the CNN extracts richer feature representations during training and simultaneously generalizes them more effectively to unseen samples. The MLP accuracy ceiling reflects the absence of spatial inductive bias rather than underfitting, as the model converges reliably but is structurally incapable of capturing the spatial relationships that define handwritten characters.

B. Training Dynamics

CNN training accuracy rises steadily from approximately 75% at epoch 1 to 95.1% at epoch 20, following a smooth learning curve consistent with stable Adam optimization. Validation accuracy tracks closely throughout training, stabilizing near 87% from epoch 14 onward. The narrow gap between training and validation accuracy of approximately 7-8 percentage points indicates that the Dropout layer successfully constrains overfitting throughout the full training duration. Loss curves on both training and validation splits exhibit smooth, monotone decay without inflection points or divergence, confirming stable convergence.

The MLP exhibits a train-validation gap of approximately 2 percentage points, suggesting mild overfitting but also a lower accuracy ceiling due to the absence of spatial inductive bias. Both models benefit from the Adam optimizer's adaptive learning rate schedule, which automatically reduces effective step sizes as gradients become smaller in later training epochs, enabling fine-grained weight adjustments that improve generalization without manual learning rate tuning.

C. Error Analysis

Inspection of the confusion matrix reveals that decimal digits (0-9) are classified with markedly higher accuracy of approximately 92-94% compared to alphabetic characters at 82-85%, owing to their lower intra-class variability and greater visual distinctiveness. The most frequent misclassifications occur between visually similar character pairs: O versus 0 with 8.2% confusion rate, I versus l with 7.9%, S versus 5 with 6.4%, C versus c with 5.8%, and Z versus 2 with 4.7%. These pairs present genuine visual ambiguity even for trained human readers.

Per-class metrics reveal that the macro-averaged F1-score across all 47 classes is 0.871. Classes with the highest recall include digit 0 at 97.1%, digit 1 at 96.3%, and digit 8 at 95.8%. Classes with the lowest recall include lowercase l at 73.2%, uppercase I at 74.8%, and uppercase O at 75.1%, reflecting fundamental ambiguity in these character pairs. Characters written with strong slant greater than 15 degrees from vertical showed approximately 3-4% lower accuracy, suggesting that slant correction as a preprocessing step could provide additional performance benefit.

Class-specific data augmentation targeting confusable pairs including controlled random stroke thickness variation, elastic distortion, and affine shear is expected to substantially reduce this error mode in future iterations. Adaptive thresholding or stroke-width normalization could address light-stroke misclassification as an additional preprocessing improvement. The confusion matrix analysis also reveals that the CNN makes qualitatively more interpretable errors than the MLP, consistently confusing visually similar characters rather than making arbitrary cross-class errors.

D. Comparative Analysis with Literature

Situating the CNN's 86.8% test accuracy within the broader EMNIST Balanced literature provides important context for evaluating the contribution of this work. Cohen et al. [9], who introduced the dataset, reported a baseline CNN accuracy of 85.5% using a shallower two-layer architecture without Dropout regularization, establishing the initial benchmark. The present CNN exceeds this baseline by 1.3 percentage points through the addition of a second convolutional-pooling stage and Dropout regularization.

Subsequent work has pushed accuracy further using more complex architectures. Attention-augmented architectures such

as CoAtNet reported by Liman et al. [10] achieved 92.3% on the Balanced split by leveraging multi-head self-attention in addition to convolutions. Residual networks fine-tuned from ImageNet weights achieve approximately 90-91% accuracy through transfer learning. The present CNN's 86.8% accuracy is achieved with only approximately 248K parameters, far fewer than ResNet-50 with 23M parameters or CoAtNet with 25M parameters.

This parameter efficiency makes the proposed model well-suited for deployment in resource-constrained environments such as embedded systems or mobile applications where computational budget is limited. The 7.6-point improvement over the MLP baseline is consistent with similar studies comparing fully connected and convolutional architectures on character recognition datasets, validating the generality of the finding that convolutional inductive bias provides substantial benefit for this class of visual pattern recognition problems.

E. Practical Deployment

The trained CNN model was serialized using TensorFlow SavedModel format and deployed as a Streamlit web application accessible through a public URL. The deployment pipeline accepts user-uploaded PNG or JPEG images of handwritten characters, preprocesses each input identically to the training pipeline by performing grayscale conversion, bicubic resizing to 28x28, per-pixel normalization, and channel reshaping, and returns the predicted character label with associated class probabilities in real time.

Sample testing with diverse user-supplied images confirmed correct predictions for all ten digit classes and 38 of 47 total classes. The nine classes with occasional incorrect predictions all correspond to the ambiguous character pairs identified in the confusion matrix analysis, suggesting that the deployment accuracy reflects the fundamental dataset ambiguity rather than model-specific failure modes. Response latency on the hosted server was approximately 80-120 milliseconds per inference, well within real-time interaction requirements for practical applications.

VI. CONCLUSION

This paper presented a systematic comparison of MLP and CNN architectures for handwritten character recognition on the EMNIST Balanced dataset encompassing 47 character classes. The CNN model, incorporating two convolutional-pooling stages, a 128-unit fully connected layer, and Dropout regularization, achieved 86.8% test accuracy and outperformed the MLP baseline by 7.6 percentage points while requiring fewer total trainable parameters of approximately 248K versus 402K. The CNN advantage is directly attributable to its convolutional inductive bias including weight sharing, local connectivity, and pooling-induced translation invariance, which aligns inherently with the spatial structure of handwritten characters.

Dropout regularization at rate 0.3 effectively controlled overfitting throughout training, producing a narrow 7-8 percentage point gap between training and validation accuracy. Confusion matrix analysis identified visually ambiguous character pairs including O/0, I/l, and S/5 as the primary source of misclassification errors. The system was further validated under real-world conditions through deployment as a real-time Streamlit web application, confirming practical applicability beyond the controlled benchmark evaluation setting. The implementation provides a fully reproducible and well-documented baseline for the EMNIST Balanced benchmark.

Future work will investigate: (1) targeted class-specific data augmentation strategies for confusable character pairs including elastic distortion and stroke-width jitter; (2) deeper residual CNN architectures with skip connections to improve gradient flow and enable greater representational depth; (3) CNN-LSTM hybrid models for word-level sequence recognition beyond isolated characters; (4) transfer learning from large-scale pretrained vision models such as VGG16 and ResNet-50; (5) application of lightweight attention mechanisms within convolutional feature maps to improve discrimination of visually similar classes; and (6) extension to multilingual and Indic script datasets including Devanagari, Telugu, and other regional Indian scripts for broader real-world applicability and societal impact.

REFERENCES

[1] R. C. Karpagalakshmi, "Deep learning-based recognition of handwritten English characters using CNN," Procedia Computer Science, vol. 229, pp. 592-600, 2025.

[2] D. Saraswathi, M. Krishnagopal, and B. S. Krishnan, "Handwritten text recognition system using machine learning and CNNs," International Journal of Computer Science, vol. 19, no. 1, pp. 308-314, Jan. 2021.

[3] S. Mahadevkar and N. Jumde, "Enhancement of handwritten text recognition using AI and CNNs," Procedia Computer Science, vol. 248, pp. 311-319, 2024.

[4] K. Bhosle and S. Walia, "Evaluation of deep learning CNN model for recognition of handwritten digits," Artificial Intelligence Advances, vol. 5, no. 1, pp. 55-63, Feb. 2023.

[5] A. K. Sharma, N. Singh, and S. Raj, "Handwritten text recognition using deep learning: A CNN-LSTM approach," International Journal of Intelligent Systems and Applications in Engineering, vol. 13, no. 2, pp. 120-127, 2025.

[6] M. A. Alshehri, A. Z. Alghamdi, and M. A. Liman, "Enhancing recognition of handwritten Arabic characters by hybrid CNN and

bidirectional RNN approaches," Sustainable Machine Intelligence Journal, vol. 9, pp. 34-56, 2024.

[7] V. V. Mukkoti and K. V. Rao, "Handwritten recognition of Telugu characters using CNN architectures," Journal of Advanced Science and Technology and Therapeutics, vol. 15, no. 3, pp. 44-60, Sep. 2025.

[8] R. S. Kumar, P. V. N. Reddy, and Y. P. D. Rao, "Deep learning-based recognition of Devanagari handwritten characters by CNN-RNN," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 7s, pp. 300-306, Jul. 2023.

[9] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 2921-2926.

[10] M. A. Liman, M. A. Rahman, and A. L. Wahab, "Handwritten character recognition using deep learning models: WaveMix-Lite and CoAtNet," Journal of Innovative Optical Health Sciences, vol. 17, no. 1, pp. 1-8, 2024.

