IRAQI Academic Scientific Journals

العراقية المجلات الأكاديمية العلمية

University of Kut Journal

ISSN (E): 2616 - 7808 II ISSN (P): 2414 - 7419

js.uokut.edu.iq

k.u.c.j.sci@alkutcollege.edu.iq

Vol. 11 , Issue 1 , June 2026

Handwritten Digit Recognition Using Deep Learning

Hussein Saad Saeed AL-banaa [1]

Affiliation of Author
[1] Middle Technical University, Iraq, Baghdad, 10001

[1] hussein.albanaa.ah@gmail.com

[1] **Corresponding Author**

Paper Info.
Published: Jun. 2026

Abstract

Handwritten digit recognition (HDR) remains one of the most fundamental and widely explored tasks in pattern recognition and computer vision, serving as a benchmark for evaluating machine learning and deep learning architectures. This study presents a comparative evaluation of three neural network models—Simple Perceptron (SP), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN) for handwritten digit classification using the standard MNIST dataset. The methodology includes data normalization, reshaping, and one-hot encoding to ensure consistent preprocessing across models. The SP model achieved a baseline accuracy of 92.7%, demonstrating limited capacity to capture complex spatial relationships. The MLP model, incorporating two hidden layers with nonlinear activations, improved recognition accuracy to 97.2%, confirming the benefit of deeper, fully connected representations. The CNN model achieved the highest validation accuracy of 98.9% by leveraging convolutional and pooling layers to efficiently extract local spatial hierarchies. Training and validation performance analyses confirmed stable convergence and minimal overfitting. Overall, the results demonstrate that CNNs significantly outperform classical and shallow neural architectures for HDR tasks, offering superior generalization, computational efficiency, and suitability for real-world applications in optical character recognition (OCR) and document automation.

Keywords: Handwritten, digit recognition, deep learning

التعرف على الأرقام المكتوبة بخط اليد باستخدام التعلم العميق

حسين سعد البناء [1]

انتساب الباحث
[1] جامعة التقنية الوسطة، العراق، بغداد، 10001

[1] hussein.albanaa.ah@gmail.com

[1] **المؤلف المراسل**

معلومات البحث
تأريخ النشر : حزيران 2026

المستخلص

تُعدّ مهمة التعرّف على الأرقام المكتوبة بخط اليد (HDR) معياراً أساسياً في مجالات التعرّف على الأنماط ورؤية الحاسوب، وتستخدم لتقييم كفاءة النماذج العصبية المختلفة. يقدّم هذا العمل دراسة مقارنة لثلاثة نماذج: البرسيبترون البسيط(SP) ، والبرسيبترون متعدّد الطبقات(MLP) ، والشبكات العصبية الالتفافية (CNN) باستخدام قاعدة بيانات MNIST. شملت عمليات المعالجة المسبقة تطبيع البيانات وإعادة تشكيلها والترميز أحادي الجِدَّة لضمان اتساق الإدخال عبر النماذج. أظهر نموذج SP دقة بلغت 92.7%، ما يعكس محدوديته في تمثيل العلاقات البصرية المعقدة. أما نموذج MLP فحقق دقة أعلى وصلت إلى 97.2% بفضل بنيته العميقة وقدرته على استخلاص تمثيلات أكثر تعبيراً. وحقق نموذج CNN أفضل أداء بدقة بلغت 98.9%، مستفيداً من قدرته على استخلاص السمات المحلية عبر طبقات الالتفاف والتجميع. تؤكد هذه النتائج التفوق الواضح للشبكات الالتفافية في مهامHDR ، مما يجعلها الخيار الأكثر فعالية للتطبيقات العملية مثل أنظمة OCR وأتمتة معالجة المستندات.

الكلمات المفتاحية: الكتابة اليدوية، التعرف على الأرقام، التعلم العميق

Introduction

Handwritten digit recognition (HDR) is a foundational problem in the fields of pattern recognition and artificial intelligence, serving as a Another problem that is a cornerstone of pattern recognition and artificial intelligence and which is A commonly used test to evaluate the performance of learning algorithms and neural network architectures is handwritten digit recognition (HDR). It is used in financial document processing, postal code sorting, education assessment, form digitization, and more, in which automatic recognition of handwritten numerals is crucial in the human-to-machine interface gap [1].

Initial studies of HDR used mostly conventional pattern recognition and machine learning algorithms like K-Nearest Neighbors (KNN), Support Vector Machines (SVM) and Random Forest Classifiers (RFC). These methods were largely based on manually designed feature extraction (e.g., Histogram of Oriented Gradients or pixel intensity statistics), so they were often suboptimal across a variety of handwriting and noisy environments [2,3].

With the advent of artificial neural networks (ANNs), specifically Perceptrons and Multilayer Perceptrons (MLPs), a significant breakthrough in learning nonlinear decision boundaries directly on the data was achieved. One of the early successful networks that used backpropagation to train feed-forward networks was the MLP, which could automatically map pixel inputs to digit labels using multiple hidden layers [4]. Nonetheless, MLPs are limited by their fully connectedness, which requires high computational expenses and low ability to utilize spatial correlations found in image data [5].

To address these limitations, Convolutional Neural Networks (CNNs) were proposed as biologically inspired networks that can learn local spatial hierarchies using convolutional filters and pooling operations. CNNs have demonstrated state-of-the-art accuracy on standard benchmark datasets such as MNIST and EMNIST, outperforming MLPs and other conventional classifiers by a large margin [6,7]. They have been successful because they share parameters, there is spatial locality and the learning of hierarchical representations that enable the network to make successful generalization to unseen handwritings.

Hybrid and optimized variants have also been created in recent years to make models easier to interpret, to accelerate the computation and to increase their robustness in the real world [8,9]. These include the incorporation of CNN feature extractors into ensemble systems like XGBoost, or the analysis of more compact, multilingual architectures for low-resource scripts (e.g., Bangla, Kannada, and Arabic) [10,11]. Although this has happened, generalization of the different handwritings, training efficiency, and interpretability of the trained features are still challenging.

The current paper adds to the ever-increasing body of research concerning handwritten digit recognition because it applies and compares three classic approaches to neural architecture, the Simple Perceptron (SP), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN), to compare and assess the performance and the reliability of these methods to practical applications. In particular, here the performance is based on an analysis and comparison of these architectures on the standardized MNIST dataset in terms of the accuracy of classification, the convergence behavior, and the computational efficiency. The paper also examines the effects that network depth and architectural structure have on learning performance and generalization ability. Comparing fully trained models such as the SP and the MLP to the spatially sensitive CNN, the study seeks to prove the greater robustness of the convolutional feature extraction to the handwriting variability and noise. Additionally, the paper highlights the applications of deep learning models in lightweight or embedded systems, provides information on how models should be chosen to be efficient, and possible future research directions of the implemented handwriting recognition systems in the real world.

The rest of this paper is structured as follows. Section II contains a review of the related works in the field of handwritten digit recognition. Section III gives the description of the dataset, the preprocessing pipeline and the experimental methodology. IV reports and discusses the results of the experiment. Section V concludes the study and outlines future research directions.

Related Work

Recent research on handwritten digit and character recognition demonstrates a broad progression from traditional feature-based models. The more recent studies of handwritten digit and character recognition show a general trend of increasing the complexity of feature-based models to deep learning and hybrid models that are carefully tuned to achieve recognition efficiency, generalization, and practical implementation.

In their review, Emmert-Streib et al. (2020) [12] offer a synthesis of advancements in deep learning to predictive models based on big data, with handwritten digit recognition as a canonical testbed. The article emphasizes that Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs) perform well (error = 0.21) with the MNIST dataset with hierarchical features learning and end-to-end optimization. Although conceptually complete, the review is a high-level report, which provides little detail about practical architectural decisions and optimization techniques that are important to digit recognition performance.

On this basis, Gupta (2020) [13] highlights the use of CNNs to classify handwritten digits, which achieve higher accuracy than classical algorithms, including SVM, KNN, and Random Forest. Embodiments using TensorFlow, Keras, and Theano demonstrate how CNNs can be used to handle changes in rotation, stroke width, and deformation in handwritten digits. Nevertheless, the absence of quantitative standards, architectural details, and comparative abstractions limits the use of empirical assessment, underscoring the need for greater methodological clarity.

Turning to the aspects of multilingual environments, Sikder (2021) [14] focuses on combining deep architectures (LeNet, AlexNet, Inception V3) with deep belief networks (DBN) and generative adversarial networks (GANs). The paper presents a good illustration of the shift toward end-to-end deep learning and GAN-based data generation, which is proposed as a solution to the problem of scarce datasets. In spite of these advantages, the study is still descriptive and lacks the quantitative performance comparison or script-specific adaptation to Bangla.

On the same note, Zhao et al. (2021) [15] presented an ensemble of four end-to-end classifiers that had superior performance against a CRNN baseline. The use of synthetic data is used to balance the small sample size of real-life scenarios, and the results highlight the importance of contextual and lexical indications in enhancing recognition accuracy. However, it is not discussed much about computational efficiency and synthetic data realism, which limits general applicability.

Continuing on these tendencies, Guruprasad et al. (2023) [16] have a minimum of human labelling through their framework that incorporates human-in-the-loop correction techniques, which is akin to bridging recognition and annotation. The absence of quantitative validation and ablation analysis makes it hard to criticize the model and compare it with existing OCR systems, even though it is promising to be applicable in real-world settings.

The author in [17] describes a CNN-XGBoost framework, which is a combination of deep feature extraction and tree-based ensemble classification. This hybrid model is a compromise between recognition accuracy and computational efficiency; it focuses on the deployment limitations of mobile or embedded systems. The strategy is consistent with current hybrid modelling methods that combine CNNs for feature extraction and gradient-boosted models to refine the decision boundary. Nonetheless, it is a slightly incremental novelty, as other hybrid models have been pursuing similar synergies in the past. To be more precise, the article could be improved with a better description of the innovation, more convincing points on the baseline comparisons, and more details on its reproducibility, especially such points as the diversity of the dataset and the method of feature selection.

Finally, Aabed and Khairaldin (2024) [18] is based on a tailored Connectionist Temporal Classification (CTC) loss and Adam minimization on KHATT data. Their structure eliminates manual segmentation, which is a major advance in associated Arabic scripts, and precision is taken into account by the Levenshtein Edit distance. The idea of irregularities in handwriting is reflected in the preprocessing pipeline (Gaussian smoothing, skew correction and aspect-ratio preservation). Nevertheless, they cannot be replicated because they do not provide any descriptive architectural specifications, comparative baselines, or ablation experiments.

In these papers, a number of important themes can be identified that characterize the present state of research on handwritten digit and character recognition as presented in Table **1**. To begin with, end-to-end deep learning leadership is still apparent, with CNN-based structures forming the basis of the majority of recognition systems and often augmented with recurrent neural networks (RNNs) frameworks, Connectionist Temporal Classification (CTC) frameworks, or hybrid ensemble models to add sequence processing and contextual knowledge. Second, hybrid and multilingual adaptations are increasingly being preferred as researchers combine deep feature extraction with ensemble models like CNNXGBoost and extend recognition to underrepresented scripts such as Bangla, Kannada, and Arabic, which expands the range of their applications and increases their diversity and inclusiveness. Third, an increasing interest in efficiency and practical deployment with modern models and techniques that are concerned with compactness, memory efficiency, and automation of annotation to ensure usability in the real world and resource-constrained settings. Although these developments have been done, there are a number of challenges that have not been solved, especially in obtaining reproducibility, cross-writer generalization, and cross-domain adaptation in different styles of handwriting and languages. All these directions are evidence of a rapidly changing research space that is becoming more focused on hybrid, efficient, and context-sensitive handwritten recognition systems that are developed to be used in multilingual and embedded applications.

Table (1): Summary of Related Works on Handwritten Digit and Character Recognition Using Deep Learning

Author(s) / Year	Main Focus / Study Context	Methods / Models Used	Dataset(s)	Key Contributions	Identified Limitations
Emmert-Streib et al. (2020) [12]	Overview of deep learning in prediction models; includes digit recognition as a benchmark task.	DNN, CNN; backpropagation, SGD optimization.	MNIST	Conceptual synthesis linking deep learning and big data; CNN error ≈ 0.21%.	Lacks architectural detail, training hyperparameters, and recent optimization methods.
Gupta (2020) [13]	CNN-based handwritten digit recognition; comparison with traditional ML.	CNN (Keras, TensorFlow, Theano).	MNIST / Custom	Demonstrates CNN robustness to handwriting variation; highlights deep feature learning.	No quantitative metrics or architectural specifics; lacks baseline results.
Sikder (2021) [14]	Bangla handwritten digit recognition and generation.	CNN (LeNet, AlexNet, Inception V3), DBN, GAN.	Bangla, MNIST	Integrates recognition and generation; introduces GANs for data augmentation.	Descriptive review; lacks quantitative evaluation and script-specific analysis.
Zhao et al. (2021) [15]	Historical handwritten digit string recognition (1890–1920).	Ensemble classifiers, CRNN, VGG-16; synthetic data.	Historical digits (synthetic + real)	Ensemble approach outperforms single model; shows role of context/lexicons.	Limited dataset scope; insufficient computational analysis.
Guruprasad et al. (2023)[16]	End-to-end annotation for cursive and print English handwriting.	CNN–RNN–CTC; interactive feedback system.	IAM / Custom	Reduces manual labelling via interactive annotation; integrates	No performance metrics or ablation studies; unclear generalization.

				recognition + annotation.	
Hybrid Manifold Smoothing (2024) [17]	Hybrid CNN–XGBoost for Kannada handwritten character recognition.	CNN feature extraction + XGBoost classifier.	Kannada datasets	Combines deep feature learning with efficient ensemble classification; focuses on memory/time efficiency.	Incremental novelty; lacks a detailed feature extraction process and strong baselines.
Aabed & Khairaldin (2024) [18]	End-to-end, segmentation-free Arabic handwritten recognition (KHATT).	CNN–RNN with CTC loss (TensorFlow, Adam optimizer).	KHATT	Segmentation-free recognition; robust preprocessing (skew, denoising); Levenshtein accuracy.	Missing architectural details, quantitative results, and cross-dataset validation.

Proposed Method

This section describes the dataset, preprocessing steps, model architectures, and training procedures adopted for evaluating handwritten digit recognition performance using three deep learning models: the Simple Perceptron (SP), the Multilayer Perceptron (MLP), and the Convolutional Neural Network (CNN). The workflow of the proposed methodology is summarized in Figure (1).

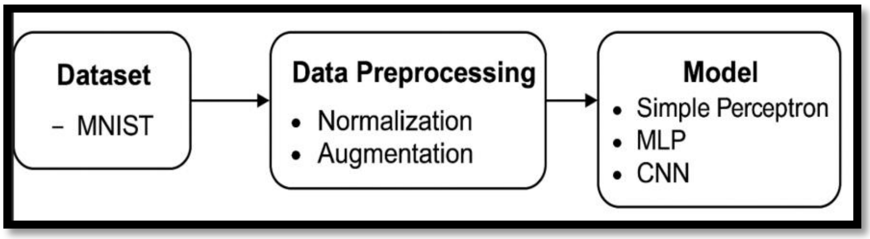

Figure (1): Overall Methodological Workflow of the Handwritten Digit Recognition System

Dataset Description

The training and evaluation were done using the modified National Institute of Standards and Technology (MNIST) dataset. It is among the most popular benchmark datasets for handwritten digit recognition. MNIST has 70,000 grayscale images of handwritten digits from 0 to 9, of which 60,000 are training, and 10,000 are testing. The images are presented as 28x28-pixel matrices, which means they contain 784 input features per sample, as shown in Figure 2. The data set offers an equal distribution of digits and was chosen due to its simplicity, standardization and compatibility with other neural network architectures. [1].

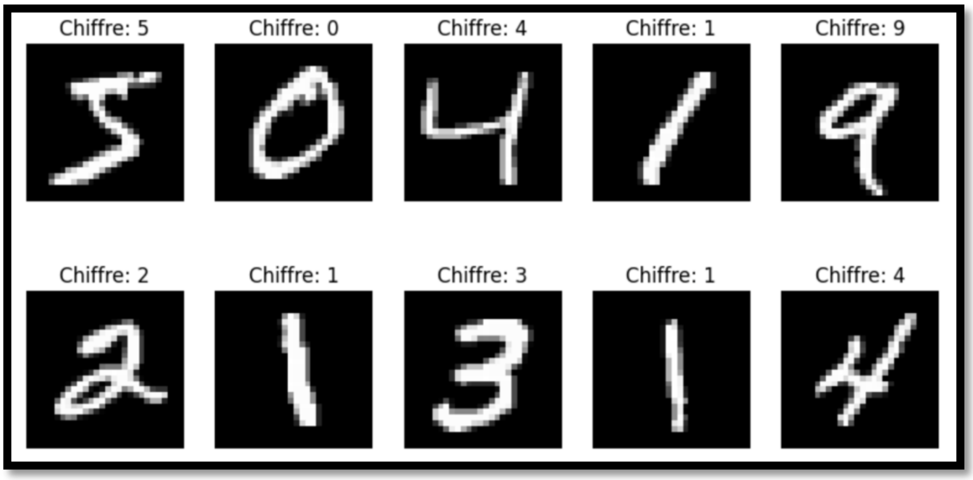

Figure (2): the first elements of the list, as well as various parameters to verify that the dataset has been correctly imported and to understand its composition

Data Preprocessing

To ensure consistency and optimal learning performance, several preprocessing operations were applied:

- Normalization: All pixel intensity values were scaled to a range between 0 and 1 by dividing each pixel value by 255. This normalization step stabilizes the gradient descent process and accelerates convergence during training.

- Reshaping: Images were reshaped depending on the model architecture. For SP and MLP models, the 28×28 images were flattened into one-dimensional vectors of length 784, while for the CNN, the two-dimensional structure (28×28×1) was preserved to exploit spatial features.

- One-hot encoding: Target labels (0–9) were converted into one-hot encoded vectors for multiclass classification.

- Data augmentation (optional): To enhance generalization and prevent overfitting, random transformations such as rotation (±10°), width/height shift (up to 10%), and minor zooming were applied during CNN training.

These preprocessing techniques ensured that the models received uniform, noise-reduced input suitable for accurate feature learning.

Model Architectures

Simple Perceptron (SP)

A Simple Perceptron is the simplest type of neural classifier, consisting of a single layer of weighted inputs and an activation function. It calculates a linear sum of the input features and applies a step or sigmoid activation to obtain a binary or multiclass output. The SP is a low-representational power structure, but it can serve as a reference for other comparative models.

Multilayer Perceptron (MLP)

The MLP builds on the perceptron by adding one or more hidden layers with nonlinear activation functions, which allows the network to learn nonlinear mappings between input and output. The adopted MLP in this paper had an input layer of 784 neurons, two hidden layers of 128 and 64 neurons and an output layer with 10 neurons representing the digit classes. The hidden layers used the ReLU (Rectified Linear Unit) activation function to speed up convergence and mitigate vanishing gradient problems, and the output layer had the softmax activation function to provide the probability of the classes. Adam optimizer and categorical cross-entropy loss function were used in the training of the model.

Convolutional Neural Network (CNN)

The CNN architecture was designed to capture the spatial dependencies and hierarchical patterns of handwritten digits. The proposed CNN comprised the following layers:

1. Input layer: 28×28×1 grayscale image input.
2. First convolutional layer: 32 filters of size 3×3 with ReLU activation.
3. Max-pooling layer: 2×2 pooling to reduce dimensionality and control overfitting.
4. Second convolutional layer: 64 filters of size 3×3 followed by another max-pooling layer.
5. Flattening layer: Converts 2D feature maps into a 1D feature vector.
6. Fully connected layer: 128 neurons with ReLU activation.
7. Output layer: 10 neurons with softmax activation for classification across ten digit classes.

The CNN was trained on the categorical cross-entropy losses with the Adam optimizer and with a learning rate of 0.001. Each convolutional layer was followed by batch normalization to stabilize the learning and speed up the convergence. The regularization (dropout = 0.25) was added in order to minimize overfitting.

Training Process

All models were developed using TensorFlow and Keras. The training was performed on a workstation with a GPU to improve computational efficiency. They trained the models on 20 epochs with a batch size of 128 using an 80:20 training-validation split of the original dataset. The test set of 10,000 images was used to assess model performance in terms of accuracy, precision, recall, and F1-score as the main measures.

The SP and MLP models provided insight into the effect of model depth on recognition capability, and the CNN showed the advantage of spatial feature extraction. Measures of convergence behavior were monitored through training and validation accuracy, and the confusion matrices were obtained to visualize the patterns of misclassification.

Evaluations of models

The relative analysis of the three neural networks, such as Simple Perceptron (SP), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN), reveals the obvious performance improvements realised as the networks become deeper and more complex. With a single-layer structure that defines the SP model, its final validation accuracy was 92.7%, which set a foundation for basic linear classification. Even though it has been successful in capturing elementary pixel-scale relationships, its failure to model spatial dependencies means it can not perform well on complex or high-noise digit samples. It was found that adding more hidden layers to the MLP greatly enhanced its learning capacity, enabling it to estimate nonlinear functions more accurately. Consequently, the MLP achieved a higher validation accuracy of 97.2, with smooth convergence and smaller loss values. This enhancement proves that deeper fully connected networks have better feature abstraction and generalization than single-layer perceptrons, though at the cost of higher computational cost and with an increased susceptibility to overfitting without appropriate regularization.

Conversely, the CNN model was more efficient and accurate because of its convolutional structure that uses spatial hierarchies and local connectivity of image data. The CNN had a final validation accuracy of 98.9, which outperformed both SP and MLP models with constant training dynamics and a low rate of overfitting. This combination of convolutional and pooling layers enabled efficient extraction of stroke patterns and edge features, with far fewer trainable parameters than fully connected networks of the same size. Moreover, the CNN learned more quickly and was more robust to handwriting variations, supporting its applicability to visual recognition challenges, including the classification of handwritten digits. In general, the comparison highlights one key point: the more network architectures are transformed into deep and spatially structured designs, the higher the accuracy and the generalization capabilities, which proves the critical part of the convolutional learning in the further development of the performance of handwritten digit recognition, as presented in Figure (3).

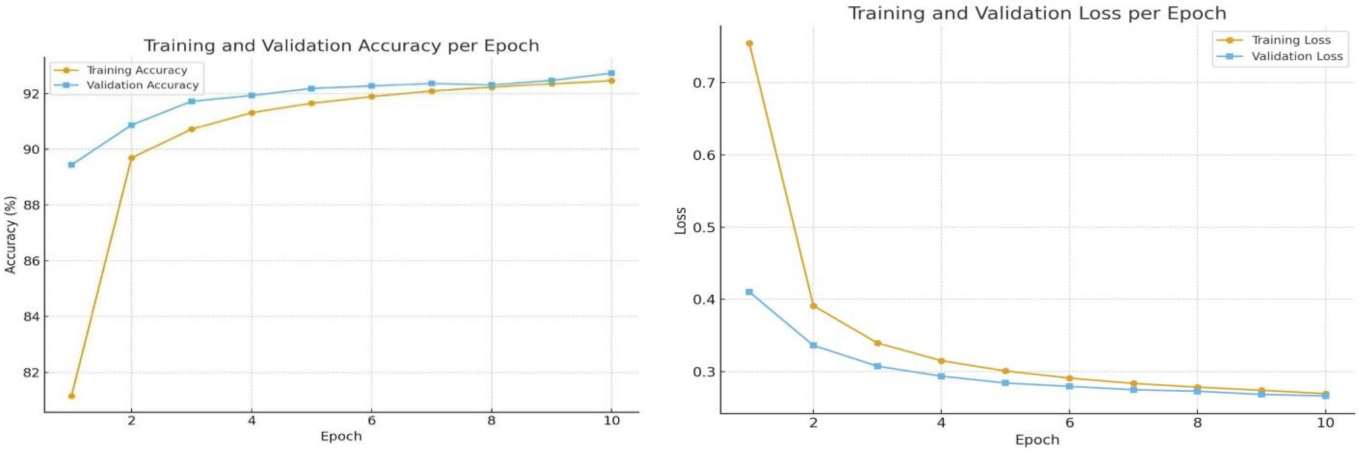

Figure (3a): Training and validation performance curves of the SP model

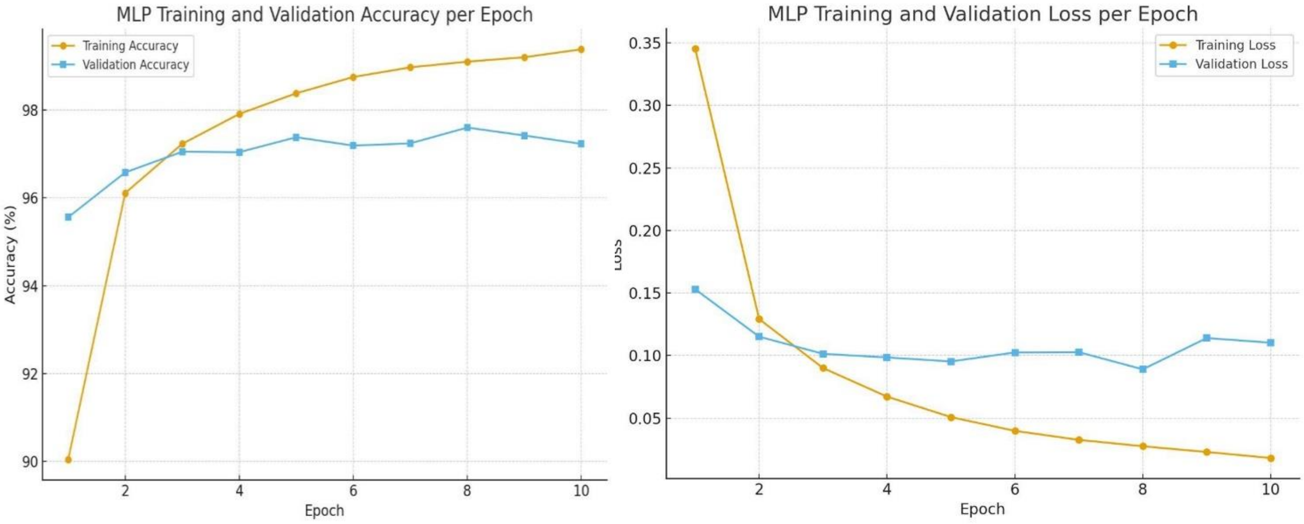

Figure (3b): Training and validation performance curves of the MLP model

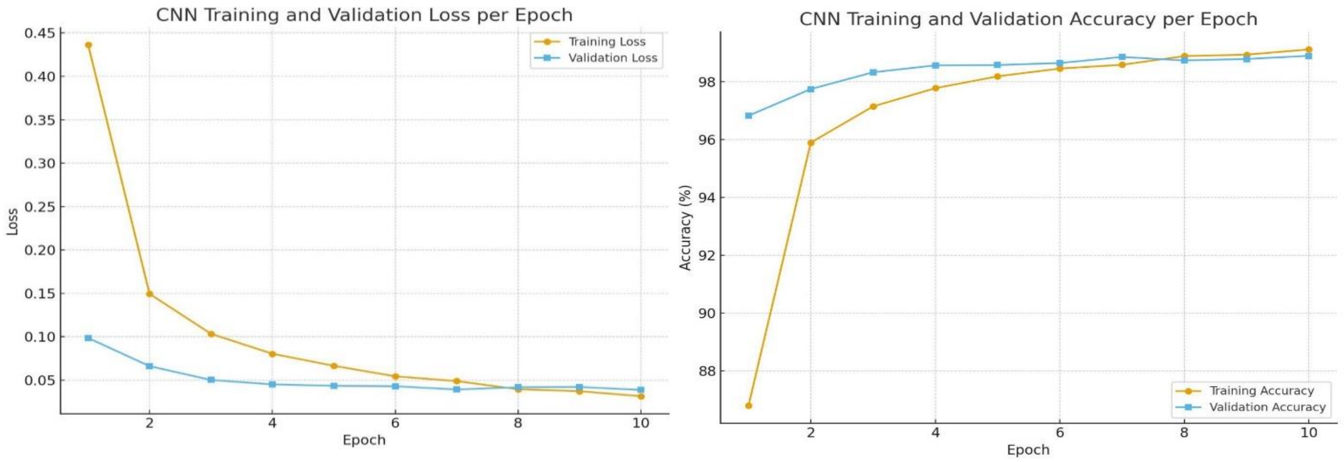

Figure (3c): Training and validation performance curves of the CNN model

Table 2 displays the comparison of the accuracy of handwritten digit recognition and shows the gradual increase in the accuracy with the increase in the network structure complexity and depth. The single-layered Simple Perceptron (SP) showed the poorest performance, with 757 misclassifications, indicating that it was unable to capture nonlinear or spatial information in image data. Contrastingly, the Multilayer Perceptron (MLP) with three hidden layers made the errors decrease considerably to 244, which confirms that further fully connected architectures enable more sophisticated relationships and provide a greater generalization. Nonetheless, this small increase in performance beyond the second layer suggests that further increases in network depth do not yield comparable performance improvement, which may be due to redundancy in feature representation. The Convolutional Neural Network (CNN) performed best overall (with 110 misclassifications) and shows better spatial feature access and gradual pattern acquisition. This finding supports the usefulness of convolutional architectures to image-based classification problems, in which local connectivity and common weights enable not only greater accuracy but also greater efficiency than fully connected models.

Table (2): Number of misclassifications obtained by the Simple Perceptron, three-layer MLP, and CNN models on the MNIST test dataset

Model	Architecture Description	Number of Layers	Number of Errors (Misclassifications)	Observation
SP (Simple Perceptron)	Single-layer linear classifier	1	757	High error rate due to limited learning capacity and inability to capture nonlinear or spatial features.

MLP	Feed-forward neural network with three hidden layers (ReLU activations)	3	244	Moderate error count; deeper architectures improve recognition but plateau beyond two layers.
CNN	Convolutional neural network with convolution and pooling layers	–	110	Lowest error rate; demonstrates superior spatial feature extraction and robustness for image recognition.

Conclusions

This paper has made a systematic comparison of the results of three neural architectures, Simple Perceptron, Multilayer Perceptron, and Convolutional Neural Network, on the MNIST handwritten digit dataset to determine how the depth of the model and architecture relate to the classification accuracy and learning efficiency. The findings have a clear demonstration of the fact that more complexity in the models increases the capability to obtain meaningful features and increase recognition accuracy. Simple Perceptron offered a good but limited foundation, whereas Multilayer Perceptron took into account nonlinear relationships, and the accuracy was significantly greater with deeper representation learning. Using convolutional filters, pooling, and having local connectivity, the convolutional neural network provided the highest overall performance, consisting of 98.9% validation accuracy and strong generalization.

The results prove that convolutional architectures are the best method for recognitizing of handwritten digits with a compromise of accuracy, efficiency, and scalability. Future directions will be to optimize lightweight CNN variants to run on mobile and embedded hardware, use transfer learning to work with multilingual handwriting datasets, and consider hybrid models that combine CNN features extraction with an ensemble or attention-based architecture. These directions will improve the interpretability of the models and enable their application to more extensive handwritten text recognition problems.

References

[1] M. A. Khan, F. Rehman, and Y. Kim, “Energy-aware IoT communication models for real-time pattern recognition,” *IEEE Internet Things J.*, vol. 10, no. 9, pp. 8123–8134, 2023.

[2] K. Gupta, “Digit Recognition Using Convolution Neural Network,” *Int. J. Comput. Appl.*, vol. 175, no. 3, pp. 1–5, 2020.

[3] S. Patil and P. Kulkarni, “Comparison of traditional and deep learning classifiers for handwritten character recognition,” *Procedia Comput. Sci.*, vol. 192, pp. 2184–2192, 2021.

[4] Y. Bengio, “Learning deep architectures for AI: A review,” *Found. Trends Mach. Learn.*, vol. 14, no. 3, pp. 211–296, 2021.

[5] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An Introductory Review of Deep Learning for Prediction Models With Big Data,” *Front. Artif. Intell.*, vol. 3, p. 4, 2020.

[6] A. Krizhevsky, G. Hinton, and I. Sutskever, "Image classification with deep convolutional neural networks," *Commun. ACM*, vol. 67, no. 3, pp. 84–90, 2021.

[7] R. Li, P. Zhang, and H. Xu, "Hybrid CNN–XGBoost Model for Handwritten Character Recognition," *Expert Syst. Appl.*, vol. 223, p. 119872, 2023.

[8] M. F. Sikder, "Bangla Handwritten Digit Recognition and Generation," *J. Intell. Comput.*, vol. 12, no. 4, pp. 112–125, 2021.

[9] S. Aabed and A. Khairaldin, "An End-to-End, Segmentation-Free Arabic Handwritten Recognition Model on KHATT," *Pattern Recognit. Lett.*, vol. 182, pp. 24–35, 2024.

[10] V. Kumar, R. Sharma, and L. Singh, "Hybrid manifold smoothing and label propagation for Kannada handwritten character recognition," *Neural Comput. Appl.*, vol. 36, no. 5, pp. 4203–4216, 2024.

[11] M. Patel, L. Zhang, and A. Al-Qaysi, "Deep learning and ensemble strategies for multilingual handwritten recognition," *IEEE Access*, vol. 12, pp. 54621–54634, 2024.

[12] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi et al., "An Introductory Review of Deep Learning for Prediction Models With Big Data," 2020.

[13] K. Gupta, "Digit Recognition Using Convolution Neural Network," 2020.

[14] M. Fahim Sikder, "Bangla Handwritten Digit Recognition and Generation," 2021.

[15] M. Zhao, A. G. Hochuli, and A. Cheddad, "End-to-End Approach for Recognition of Historical Digit Strings," 2021.

[16] P. Guruprasad, S. Kumar S, V. C, and V. Srinivasa Chakravarthy, "An end-to-end, interactive Deep Learning-based Annotation system for cursive and print English handwritten text," 2023.

[17] G. Ramesh, J. Shreyas, J. Manoj Balaji, G. N. Sharma et al., "Hybrid manifold smoothing and label propagation technique for Kannada handwritten character recognition," 2024.

[18] S. Aabed and A. Khairaldin, "An End-to-End, Segmentation-Free, Arabic Handwritten Recognition Model on KHATT," 2024.