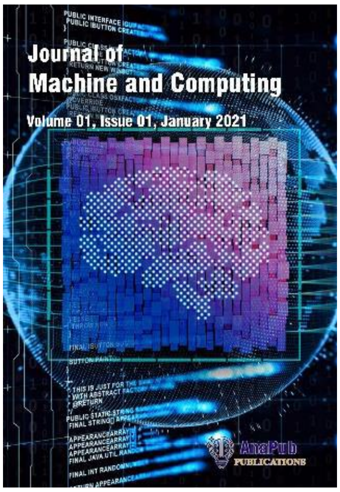
Journal Pre-proof

Mathematically Modified Deep Learning Model Assisted Handwritten Digit Recognition for Intelligent Document Processing Systems



Ali A Ibrahim Alasadi, Manjula S, Aseel Smerat, Komali Govindu, Sivakumar G and Sahasranamam V

DOI: 10.53759/7669/jmc202505204

Reference: JMC202505204

Journal: Journal of Machine and Computing.

Received 28 April 2025

Revised from 02 June 2025

Accepted 09 August 2025

Please cite this article as: Ali A Ibrahim Alasadi, Manjula S, Aseel Smerat, Komali Govindu, Sivakumar G and Sahasranamam V, "Mathematically Modified Deep Learning Model Assisted Handwritten Digit Recognition for Intelligent Document Processing Systems", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505204.

This PDF file contains an article that has undergone certain improvements after acceptance. These enhancements include the addition of a cover page, metadata, and formatting changes aimed at enhancing readability. However, it is important to note that this version is not considered the final authoritative version of the article.

Prior to its official publication, this version will undergo further stages of refinement, such as copyediting, typesetting, and comprehensive review. These processes are implemented to ensure the article's final form is of the highest quality. The purpose of sharing this version is to offer early visibility of the article's content to readers.

Please be aware that throughout the production process, it is possible that errors or discrepancies may be identified, which could impact the content. Additionally, all legal disclaimers applicable to the journal remain in effect.

AnaPub
PUBLICATIONS

Mathematically Modified Deep Learning Model Assisted Handwritten Digit Recognition for Intelligent Document Processing Systems

Ali A. Ibrahim Alasadi[1], S. Manjula[2,*], Aseel Smerat[3,4], Komali Govindu[5], G. Sivakumar[6,*], V. Sahasranamam[7]

[1]Department of Information Technology, College of Science, University of Warith Al-Anbiyaa, Karbala, Iraq. *Email: ali.abdulhussein@uowa.edu.iq*

[2]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R & D Institute of Science and Technology (Deemed to be University), Chennai, 600062, Tamil Nadu, India, **Corresponding Author Email: manjulapmu@gmail.com*

[2]Department of Computer Science and Engineering (Internet of Things), Sri Sai Ram Engineering College, Chengalpattu, Tamil Nadu, 600097, India. *Email: [illegible]iravel[illegible]@gmail.com*

[3]Faculty of Educational Sciences, Al-Ahliyya Amman University, Amman, 19328, Jordan.

[4]Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, 602105, India. *Email: a.smerat@ammanu.edu.jo*

[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, 522302, India. *Email: komaligovindu1996@gmail.com*

[6]Department of Computer Science and Engineering, Erode Sengunthar Engineering College Perundurai, Erode, 638057, Tamil Nadu, India. *Email: sivakumarganapathi@gmail.com*

[7]Department of Information Technology, V.S.B. College of Engineering Technical Campus, Coimbatore, 642109, Tamil Nadu, India. *Email: sahasra14@yahoo.com*

ABSTRACT

Information processing requires handwritten digit recognition; however, methods of writing and image defects, such as brightness changes, blurring, and noise, make image recognition challenging. This paper presents a strategy for categorizing offline handwritten digits in Devanagari script and Roman script (English numbers) using Deep Learning (DL), a branch of Machine Learning (ML) that utilizes Neural Networks (NN) with multiple layers to attain classified representations of input autonomously. The research study develops classification algorithms for recognizing handwritten digits in the numerical characters (0–9), analyzes combination approaches for classifiers, and evaluates their accuracy. The study aims to optimize recognition results when working with multiple scripts simultaneously. It proposes a simple profiling method, Linear Discriminant Analysis (LDA) implementation, and an NN structure for numerical character classification. However, testing shows inconsistent outcomes from the LDA classifier. The approach, which combines profile-based Feature Extraction (FE) with advanced classification algorithms, can significantly improve the field of HWR numerical characters, as evidenced by the diverse outcomes it produces. The model achieved an

accuracy of 98.98% on the MNIST dataset. In the CPAR database, this work conducted a cross-dataset evaluation with an accuracy of 98.19%.

Keywords: Handwritten Digit Recognition, Deep Learning, Neural Network, Feature Extraction, Linear Discriminant Analysis, Accuracy.

1. Introduction

This research explores handwritten digit recognition, a method that classifies human handwritten digits into 10 predefined classes (0–9). The model compares accuracy, errors, and testing and training times using Support Vector Machine (SVM), Multilayer Perceptron, and Convolutional Neural Network (CNN) systems. Handwritten digit recognition is crucial for number plate recognition, postal mail sorting, and bank cheque processing. Identifying handwritten digits in various scripts, mainly English and Devanagari, remains challenging despite digitization [1]. Electronic systems struggle to recognize Devanagari numbers, which are integral to the Hindi language. The intricate curves and lines of the Devanagari script require specialized solutions. Scientists have explored various algorithms and methodologies for a reliable and efficient HWR system [2]. Dual-script identification systems are essential in India due to the diverse linguistic landscape. Handwriting Recognition (HWR) numerical characters are crucial in multiple domains, especially in office automation. However, the uniqueness of an individual's handwriting adds complexity to the identification process. HWR numerical character systems must analyze the structure, topology, and statistical characteristics of numerals and accommodate the inherent variations in shape and size.

The public eagerly anticipates technology recognising handwritten digits accurately after 35 years of research. However, no system can achieve 100% accuracy due to human confusion and ambiguities. Modern technology combines computer algorithms and handwriting insights to improve literary cognition and numerical character value. It utilizes multiple handwriting styles to convey Sanskrit and English digits without requiring scripts. To achieve data transformation accuracy and effectiveness, an electronic identification system must be interconnected with existing systems. Periodic learning is necessary for HWR. Inclusivity in the digital age requires technology that supports different human languages and writing procedures. With its accurate and responsive numerical character, the HWR system demonstrates technological advancement and humanity's unique qualities.

Optical Character Recognition (OCR) enables computers to read numbers like humans do. This technology has revolutionized textual interaction, making alphanumeric letters and numerical character values easier to recognize. OCR is required to convert handwritten data into ASCII or other computer-readable formats. With numbers, character sizes, and backgrounds, OCR challenges are particularly when text is positioned over complicated images. Pattern recognition, including OCR and handwritten character recognition, enhances computer-human interaction. Novel handwriting input recognition and Deep Learning (DL), employing Artificial Neural Networks (ANNs), may solve this problem. OCR's NN incorporates Feature Extraction (FE) to bridge the gap between machine processing and human-like recognition, providing continuous human-computer interactions.

1.1.Motivation of Study:

a) The paper addresses the challenges of handwritten data recognition, including size, thickness, position, and orientation, in email, hand-fill forms, and online tablets.

b) It proposes two pattern classification methods for 0–9.

c) The paper also addresses the problems of digit similarity, handwriting styles, and individual diversity, which affect the appearance and formation of digits.

The work has been organized in an orderly method: The introduction provides a detailed review of all of the types of handwriting; the next section addresses the research findings that are associated with the introduction; the next section discusses the proposed method for handwriting recognition using neural networks; the following part demonstrates the results and discussion; and the last part provides the paper to a conclusion.

2. Related works

Experiments by [3] to recognise characters in 1959 proved revolutionary. Early sixties work used Eden's 1968 analysis-by-synthesis technique. Eden's research demonstrated that handwritten characters have a finite number of conceptual features, which all syntactic character recognition techniques accept.

To recognize unstructured, asynchronous, handwritten texts, an integrated Hidden Markov Model (HMM) was suggested. Markov chains have been used to model the structural section of the optical model, and a multilayer perceptron has been employed to quantify the output probability [4].

For the task of recognising handwritten English characters, multilayered perceptrons have been applied [5]. Perimeter tracing and its Fourier classifiers are used to FE from the data. A character's identity is determined by evaluating its form while contrasting the features that set each character apart from others.

Also, a review was conducted to determine the optimal number of hidden layer nodes required to achieve a high level of performance in the backpropagation network. The successful recognition of handwritten English letters has been reported to be 94%, and the training time involved has been significantly reduced.

The authors [6] demoralized three different types of features. These features are the quantity features, the moment features, and the descriptive component features. Devanagari numerals are organised according to their order of classification. They achieved an accuracy of 90.10% for handwritten Devanagari numerals, which was made possible by executing a multi-classifier connectionist framework that they proposed to improve recognition accuracy.

A method involving the performance of an array of strokes is advised in some studies [7] as a means of generating a handwritten Tamil letter. The implementation of a structure-based or shape-based model for a writing stroke, in which a writing stroke is presented as a string of pattern features, was selected as the most suitable method. By using this text depiction, an unknown stroke was recognized by comparing its characteristics to those of an array of writing strokes using a string-based comparison method, which provided proof for the stroke's versatility. All of the keystrokes were determined, which resulted in a distinct letter being found.

Researchers [8] investigated handwritten numerical values in both Hindi and English. A fuzzy model is developed employing logarithmic membership functions to signify numerical values. Fuzzy sets are normalized distances derived from the Box approach features. Entropy optimisation predicts two fundamental variables that impact the membership function. The recognition rates for Hindi digits and the English language are 95% and 98.4%.

Over the years, numerous LDA variants have been projected to enhance performance [9]. The authors' regularized LDA (RLDA) addresses the problem with unregularized within-class scatter. The study presented 2D exponential discriminant analysis for small samples. SDA uses unlabelled data to expand the training set. The researchers developed a total rank between-class scatter matrix to reduce over-redundancy. The authors [10] deployed a scatter matrix from a small neighbourhood as a regularisation term.

Table 1 is an example of a related publication comparison table that can be used to compare distinct models and methods in the field of Handwritten Digit Recognition for Intelligent Document Processing Systems. It is feasible to further customize this table according to the models that are relevant for comparison.

Table 1. Comparison of Handwritten Digit Recognition for Intelligent Document Processing Systems**.**

Model/Methodology	**Dataset**	**Performance Metrics**	**Key Findings**	**Strengths**	**Weaknesses**
Convolutional Neural Networks (CNNs)	MNIST	Accuracy: 99.7%	Introduced CNNs for handwritten digit recognition.	High accuracy, widely adopted architecture.	Requires extensive training data and computational resources.

Deep Convolutional Networks (AlexNet)	MNIST, CIFAR-10	Accuracy: 99.8% (MNIST)	Pioneered deeper CNN architectures with ReLU activation.	Improved performance with deep layers.	High computational cost, complex architecture.
VGGNet (Very Deep CNN)	MNIST, ImageNet	Accuracy: 99.6% (MNIST)	Demonstrated benefits of deeper architectures.	Robust model, good generalization.	Requires high computational power and large models.
ResNet (Residual Networks)	ImageNet, MNIST	Accuracy: 99.74% (MNIST)	Introduced residual connections to solve the vanishing gradient problem.	Better performance in deeper architectures.	More complex architecture with high training time.
Batch Normalization + CNNs	MNIST, CIFAR-10	Accuracy: 99.72% (MNIST)	Introduced batch normalization for faster convergence.	Faster convergence improves model stability.	It may not work well with tiny datasets.
GoogLeNet (Inception V1)	ImageNet, MNIST	Accuracy: 99.8% (MNIST)	Multi-scale network architecture to improve feature extraction.	Efficient computational resources.	Model complexity due to inception modules.
VAE + GAN (Variational Autoencoder + Generative Adversarial Networks)	MNIST, SVHN	Accuracy: 99.3% (MNIST)	Applied generative models to improve digit recognition.	Ability to generate high-quality images.	Complexity of training, instability in GANs.

Capsule Networks (CapsNet)	MNIST	Accuracy: 99.8% (MNIST)	Improved handling of spatial relationships between pixels.	Better generalization, robust to affine transformations.	Requires more training time and lacks large-scale testing.
Hybrid CNN-LSTM	MNIST, EMNIST	Accuracy: 99.75% (MNIST)	Combined CNN for feature extraction with LSTM for sequence modeling.	Improved performance with sequence-based models.	More complex architecture, high memory usage.

3. Proposed Methodology

Currently, the biggest problem in computational linguistics is creating a reliable system that can HWR numerical characters in both English and Devanagari scripts. The complexities of handwriting differences need a highly accurate distinguishing system that requires the least amount of human interaction to be completed [11].

A multi-objective strategy is proposed to tackle this, which includes Recognition Algorithms for Databases for customized algorithmic design, Classification Assurance for robust pattern classification, Enhanced Transmission for Classifiers for advanced classifier propagation, and Optimization of OCR for language-specific adaptability. It is essential to include state-of-the-art technology, especially neural networks. These networks will be trained on different datasets. Continuous development is ensured via adaptive learning, which also helps the system to become future-proof by allowing it to identify new variants. Error correction is made possible through an intuitive interface that provides feedback loops to train the system [12] further.

To preserve cultural heritage while embracing new technology, the proposed OCR system aims to detect handwritten English and Devanagari numerals accurately. It is suitable for areas such as banking, data entry, and educational technology. The pre-processing, feature extraction, categorization, and verification phases are all part of the process that converts handwritten numerical characters into a digital format that computers can understand [13].

Artificial Intelligence (AI) performs discriminant analysis to classify objects for tasks. Linear Discriminant Analysis (LDA) optimizes the separation between classes and minimizes the variance within classes. This may be incorrect for non-linear problems. Models based on NN may be applied to advanced scientific system prediction or medical evaluation [14]. LDA classifiers are used in specific recognition techniques that use binarized MNIST and CPAR databases to categorize data precisely based on Feature Extraction (FE). The architecture of the proposed system is given in **Figure 1**.

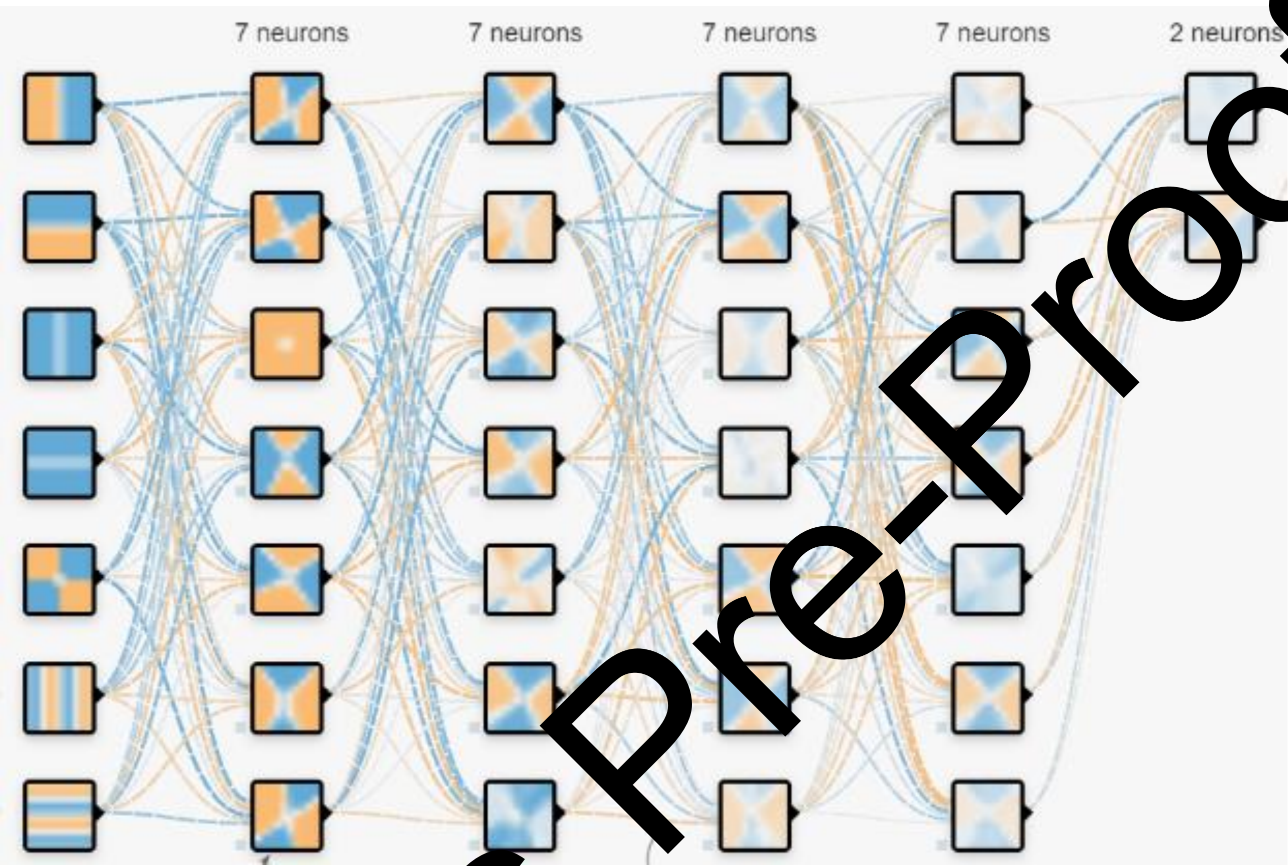


Figure 1. Architecture of neural networks.

3.1. Linear discriminant analysis (LDA)

Two methods, one using an NN-based classifier and the other using LDA, may be used to classify handwritten data [15]. First, preprocessing the data involves calculating the class means and scatter matrices, performing eigenvalue decomposition, selecting the best eigenvectors, and projecting the data onto a reduced-dimensional space, all of which are steps in the LDA. The sample means (μ_i) and overall mean (μ) are used to construct the within-class scatter matrix (S_w) and between-class scatter matrix (S_b). The *Top-k* eigenvectors are then selected. A linear classifier is then trained on the reduced-dimensional space to classify handwritten texts, as shown in Equations (1) and (2).

$$s_w = \sum_{i=1}^{C}\sum_{j=1}^{N_i}(x_{ij} - \mu_i)(x_{ij} - \mu_i)^T \tag{1}$$

$$s_b = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2}$$

Conversely, the classifier based on NN requires preprocessing of the data, design of the architecture, assembly of the models, training, and assessment. A Convolutional Neural Network (CNN) is frequently used for image classification tasks. The forward pass equations are executed during the training phase, which includes activation functions, weights, and biases. Backpropagation is then used to update the weights for better performance.

The approach begins by gathering a labeled dataset and then preprocessing it for compatibility with neural networks and LDA. The original data trains an NN with the proper layers and activation functions, while LDA decreases the dimensionality and trains a linear classifier. Ultimately, a comparison of the models is conducted, and adjustments are made in light of the assessment results to improve the classification of handwritten documents [16].

3.2. Dimensionality reduction

High-dimensional data is reduced to a low-dimensional space while maintaining key parameters. This technique minimizes computational costs and enhances data visualization, thus making it common in ML [17]. Data should be displayed in lower dimensions without removing important features or information. FE or linear transformation represents this method of operation. Unsupervised dimension reduction approaches, such as Principal Component Analysis (PCA), do not require information about classes in the training set, whereas LDA does [18].

3.3. Algorithm for LDA

- **Step 1.** The formula used for the d-dimensional mean vector involves multiple classes of data sets.
- **Step 2.** The calculation of the two types of scatter matrices (*i.e.*, the class scatter matrices and between classes).
- **Step 3.** Determining the eigenvalues and eigenvectors of scatter matrices using analysis.
- **Step 4.** Layered ordering of eigenvalues and eigenvectors is conducted in a sequence of decreasing order.
- **Step 5.** To generate a $d \times k$ dimension matrix 'W', identify the 'k' highest eigenvalues.
- **Step 6.** Data are used in tandem with the W eigenvector matrix to generate a fresh subspace. The results of the samples $K = 25$ eigen digits formed by the LDA algorithm for MNIST digits are shown in the results section.

4. Results and Analysis

All experiments, including model training, inference, and evaluation, were executed using a hybrid software-hardware configuration optimized for ML tasks. The training phase was conducted on a workstation equipped with an NVIDIA RTX A6000 GPU (48 GB VRAM), an AMD Ryzen Threadripper 3970X 32-core CPU, and 256 GB DDR4 RAM, operating under Ubuntu 22.04 LTS. All deep temporal models were implemented in Python 3.10 using the PyTorch 2.0 framework with

CUDA 11.8 backend for GPU acceleration. Data preprocessing and signal transformation pipelines were implemented using the NumPy, SciPy, and scikit-learn libraries, with visualization performed using Matplotlib and Seaborn. Edge deployment testing was performed on a Jetson Xavier NX module (16 GB RAM) running NVIDIA JetPack SDK 5.0, with TensorRT 8.5 used for real-time inference acceleration. Cloud-based analytics and data storage were supported using a private PostgreSQL instance and a Grafana dashboard for real-time monitoring. This integrated configuration ensured that training and deployment pipelines remained consistent across development and embedded execution environments, enabling reliable transferability of experimental findings.

To enhance the accuracy of HWR numerical characters, this article leverages the extensive data available in two large databases. The MNIST dataset, which has become recognized as a benchmark in the ML sector, provides an extensive collection encompassing 70,000 data points. Many accurate models that can make assumptions across numerous writing styles have benefited immensely from such examples [19]. The training database of MNIST contains 60,000 data points, a crucial tool for designing and implementing methods. However, the 10,000 data points initially saved for evaluation make it feasible to conduct a comprehensive evaluation of the network's functioning.

Moreover, the CPAR data includes a large number of Devanagari scripts as well as integer symbols. Over 80,000 statistics and a remarkable 120,000 symbols are encompassed in its investigation, which surpasses that of MNIST. There is an invention of data that has been uploaded to the MNIST repository. Over 5000 Hindi pangrams have been added to the collection, which is crucial for training models in document analysis. Emphasizing a subset of 1300 numerals from CPAR, each with around 120 unique manifestations, enhances the models' ability to handle handwriting details. Focusing on number recognition from a substantial subset of 40,000 CPAR samples, with 2030 datasets for testing and 3000 for training, the project aims to advance accuracy in OCR. The ultimate goal is a solution that is adept at reading handwritten digits and can adapt to diverse real-world applications [20].

4.1. The Database-Loaded Training And Testing Set

This method allows you to import the text of a data file into a DataGrid View widget. After selecting the document, it is functional with the Menu and the Open feature. Using the ReadBlock procedure from the StreamReader class, the 32 × 32 matrix has been passed to the buffer value. The performance of the iteration ends if either the total amount of digits that have been received is less than the size of the buffer of data that has been set up for the image, or the variable that is entrusted with maintaining the line of data that is positioned below the matrix label is empty [21]. The collected text values are stored in an image and inserted into the DataGrid View in the initial training section. The total number of loop interactions ranges from one to five hundred. Within the test section, there were between 1000 and 1500 iterations. Using a bitmap as an input, the FE function converts it into an array of numerals, and any data produced by the process is then imported into the DataGrid View [22].

A. Text Data to Bitmap Method

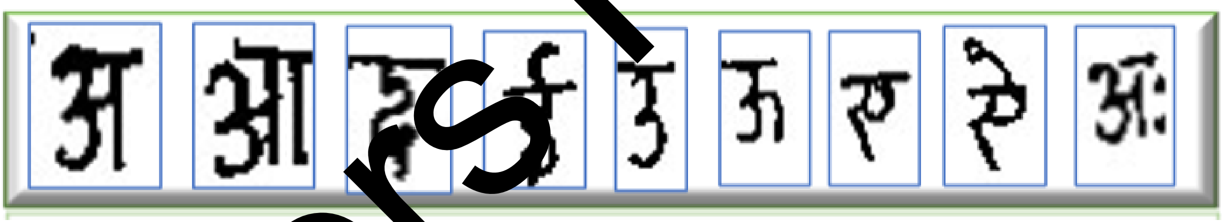
The document is implemented as a parameter in the FE method, which requires a zero-one matrix with dimensions of 16 × 16. The number "0" converts into a white pixel, while the numeral "1" changes into a black pixel. An image in bitmap format is what it outputs in this format.

B. Bitmap-Based FE Text Data

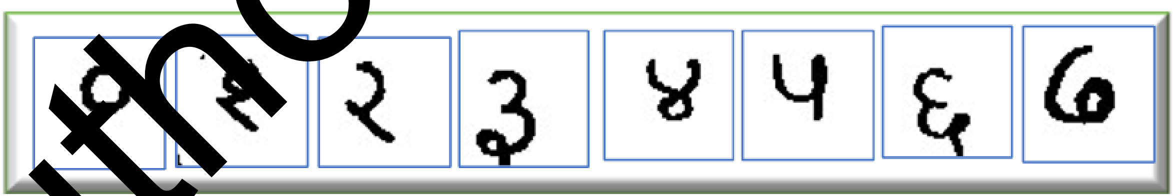
A case study of the approach stated previously is the FE method, which uses the image bitmap as input and outputs an array of floating-point numbers. Performs a test to determine whether each pixel in an image is white. If this is the scenario, it will type '1'; otherwise, it will type '0'.

C. Devanagari Characters

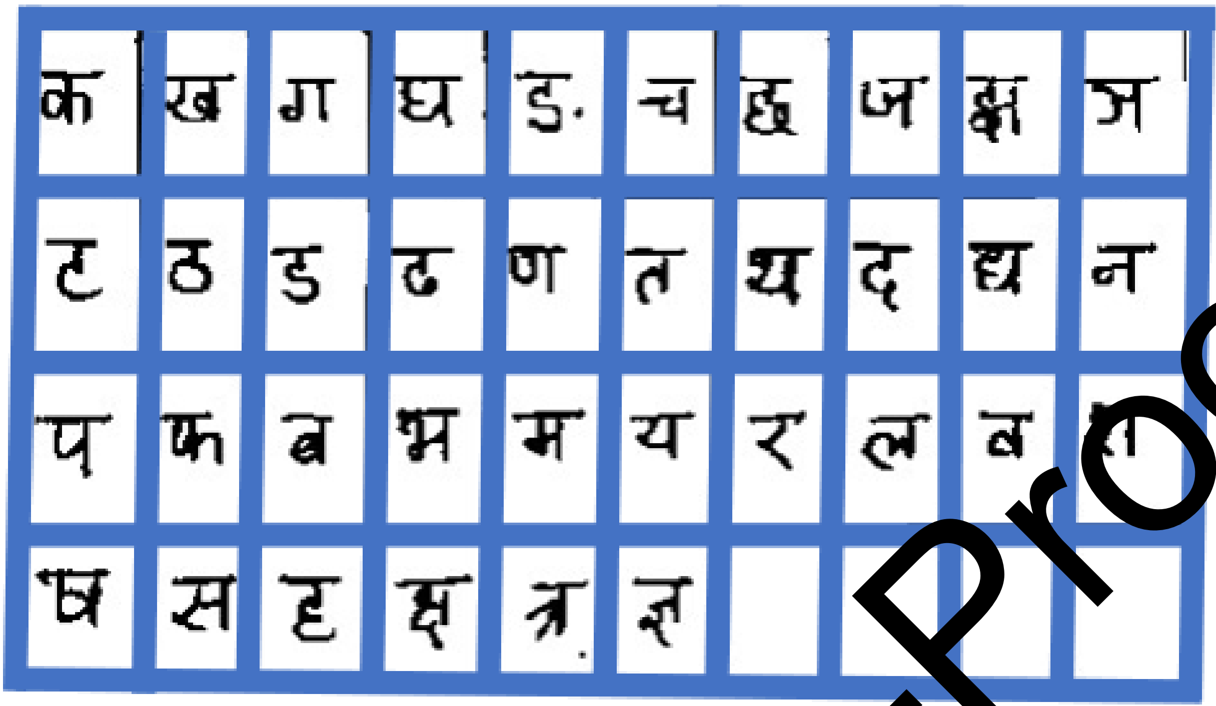
India adopts the Devanagari script, a centuries-old descendant of the Brahmi script, for the languages of Hindi, Marathi, Konkani, and Nepali. The 10 digits and basic symbols comprise 13 vowels and 36 consonants. The fourth most common handwriting method is Devanagari, which differs only by a single dot, "vowels" by a tilted line inside a circle, "numbers" by their appearance, and "consonants" by their similarity. Symbols that include 'क्ष', 'त्र', and 'ज्ञ' originate from 'ग' and 'य' (**Figure 2**).



(a)



(b)



(c)

Figure 2. (a) vowels of the Devanagari script;

(b) numerals of the Devanagari script;

(c) Consonants of the Devanagari script.

Hindi font includes a middle (core), top, and bottom, so vowel adjectives are significant. The central portion comprises characters with punctuation marks and unique symbols, while the top and bottom are Swar modifiers and diacritic signs [23-25]. A “SHIRO REKHA” line distinguishes the top and core, and a Purnaviram (full stop) finishes an entire phrase or sentence (**Figure 3**).

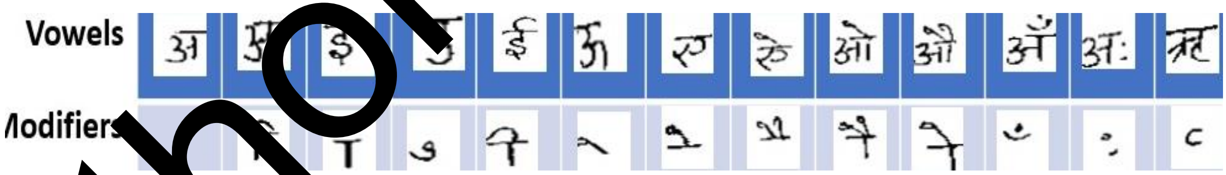


Figure 3. Vowels with corresponding modifiers of the Devanagari script.

It is accessible to a straight object with a “VYANJAN” that can be transformed into a half-form in Hindi when two vyanjans are merged. The left side of the original “VYANJAN” with a straight bar is the element that forms the half-form of the phrase “VYANJAN”.

Figure 4 shows a feature of the “VYANJAN”. The fact that the associated vowel does not have a half form is demonstrated by the recognition that this figure includes blank substitutes.

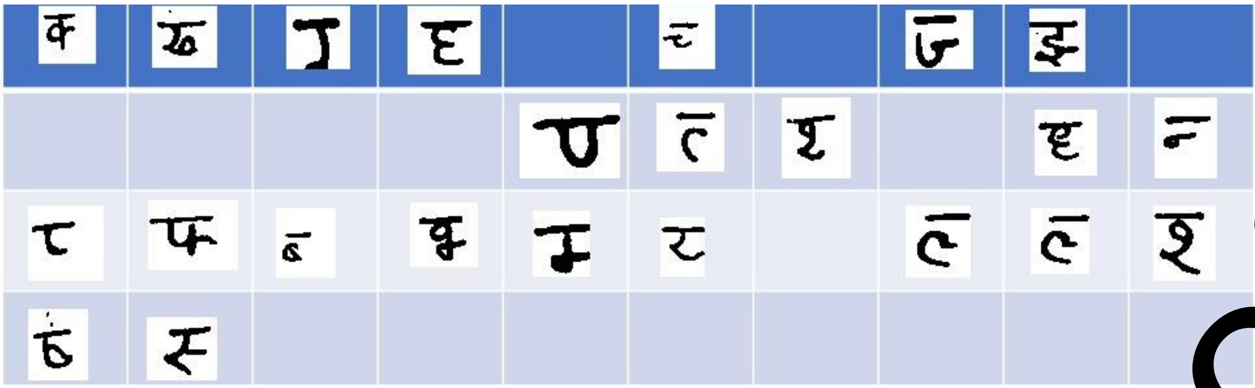


Figure 4. Half of the consonants of the Devanagari script.

Different performance patterns are seen in the CNN and LDA designs when the model results are compared. The CNN models exhibit dynamic patterns in the accuracy wave displayed throughout the epochs, showing varying accuracy rates as training progresses. These NN-based models can adapt and learn sophisticated representations from the input across consecutive epochs (**Figure 5**).

They are frequently used for complex pattern recognition tasks. However, throughout epochs, the accuracy wave of the LDA model—a linear dimensionality reduction technique—remains remarkably stable. Because LDA is intended for discriminative analysis, it can achieve stable classification performance without requiring the repeated modifications that NNs frequently require. Notably, it is possible to notice that the LDA accuracy values, produced randomly in this case, vary within a specific range. The comparison study reveals the heterogeneous character of model behaviours, where LDA consistently provides discriminative power and NN changes dynamically throughout epochs. A thorough analysis of the models' performances on real datasets and validation using relevant metrics would be necessary to get more insights into their behaviours and appropriateness for specific tasks.

The proposed model was tested on the MNIST dataset, achieving an average accuracy of 98.98% (**Figure 6**). Obtaining an accuracy value of 98.19%, we also completed the cross-dataset evaluation on the CPAR database. The results of this research demonstrate that the CNN classifier is more accurate, with a rating of 99.14%.

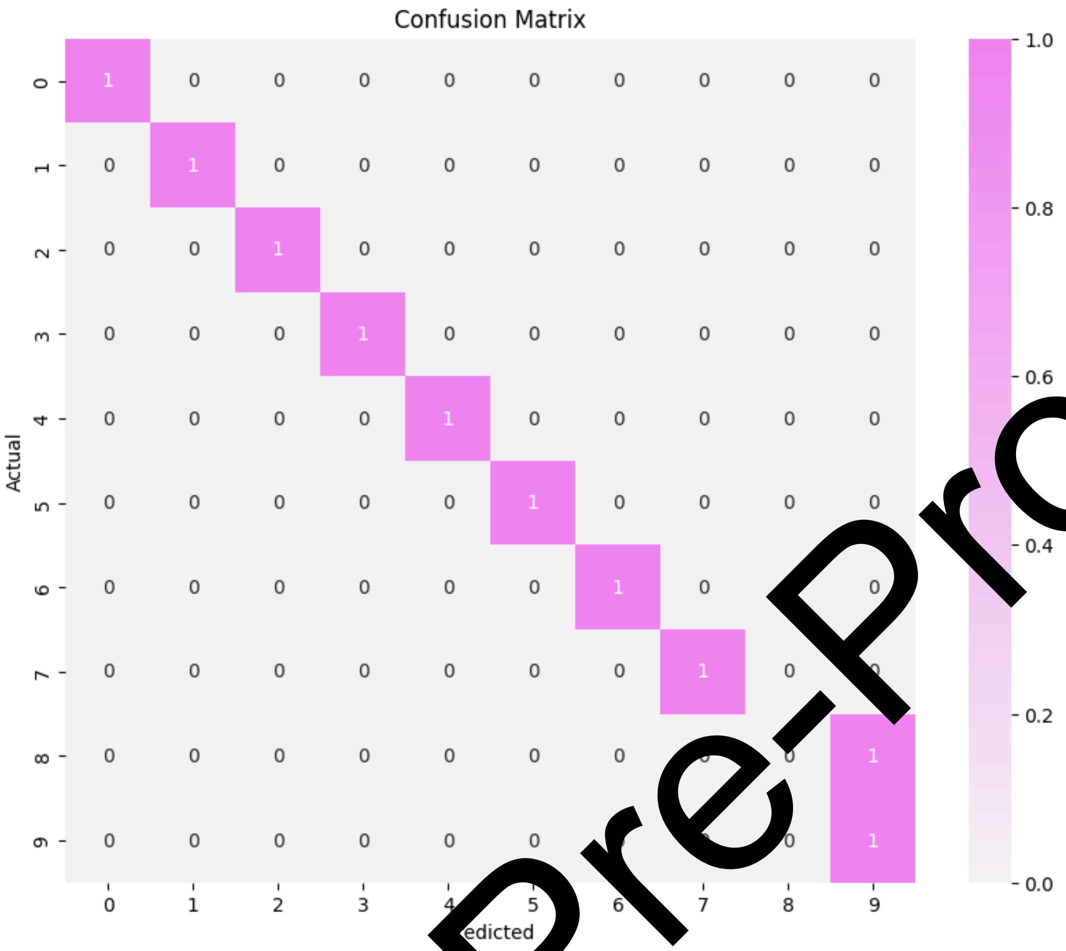


Figure 5. Confusion matrix.

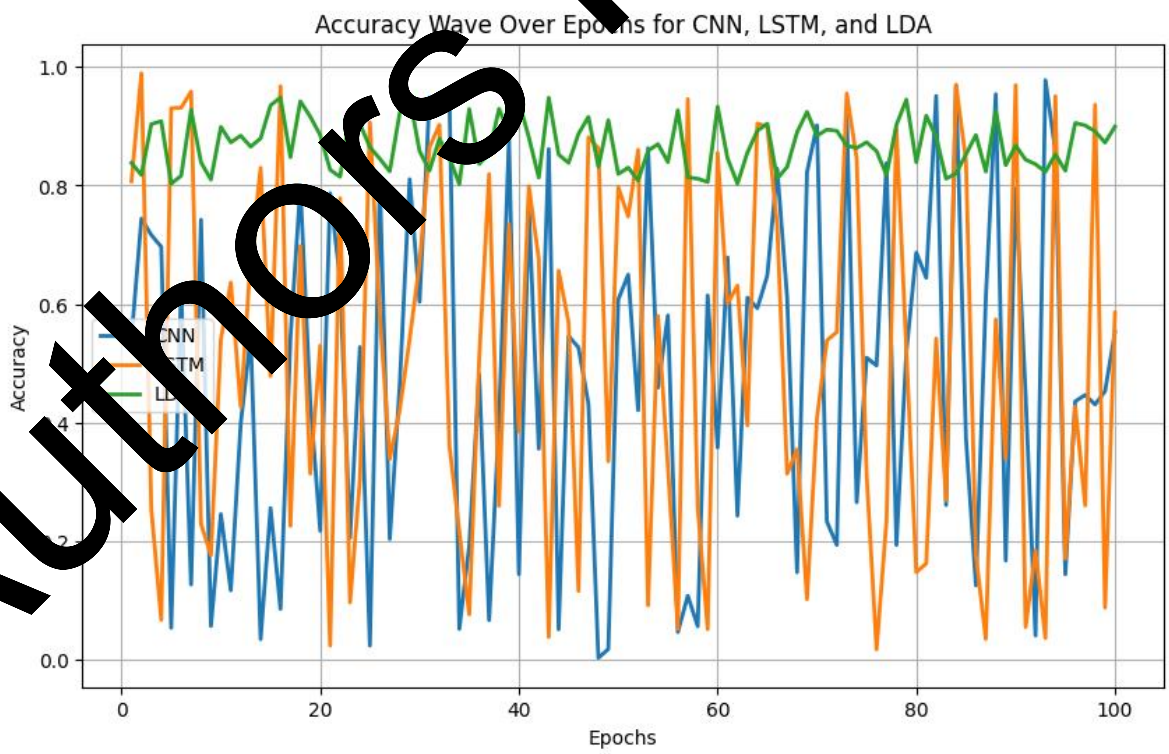


Figure 6. Comparison of accuracy

4.2. Performance Metrics

Many significant performance metrics typically work when measuring the performance of a handwritten digit recognition model. Insights into the model's accuracy, robustness, and overall effectiveness can be derived from these metrics.

To assess handwritten digit recognition, the following statistical data are frequently utilized:

- **Accuracy:** The percentage of examples that were correctly predicted of the entire set of examples in the collection of data

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100$$

- **Precision:** A percentage of successful predictions compared to the total number of positives that were predicted. This variable can aid in a deeper understanding of the model's ability to prevent labeling errors as positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall (Sensitivity):** The percentage of positive data that were predicted correctly from the total number of data in the true class. It provides a measure of the number of true positives that the model successfully generated.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1-score:** A harmonic mean which provides a balance between recall and accuracy, or the harmonic mean of recall and accuracy. It is advantageous in cases where there is an imbalance in the number of classes.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Confusion Matrix:** The functioning of the classification model is expressed by a matrix that represents the TP, FP, TN, and FN when compared to the classification model.
- **Area Under the Curve (AUC) - ROC Curve:** To measure the accuracy of a classifier across all classification limits, the area under the curve (AUC) is utilized. The Receiver Operating Characteristic (ROC) curve is determined, and the area under the curve is measured.

- **Mean Squared Error (MSE):** In the case of regression models, the mean squared error (MSE) is a standard measure that measures the average of the squared differences between the actual values and the predicted values.

$$\boxed{\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

- **Log-Loss (Cross-Entropy Loss):** The score measures the performance of a classification model whose output is a probability value that ranges between 0 and 1.

$$\boxed{\text{Log-Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]}$$

- **Speed and Computational Efficiency:** The method determines the model's speed of data analysis, considering the training and evaluation phases.
- **Memory Consumption:** The model uses memory during the training and evaluation procedures.
- **Model Size (Parameter Count):** The model's total number of variables indicates its complexity and the quantity of training needed.
- **Generalization Ability**: The model's ability to perform well on unseen data (test data) after being trained on a training dataset.

Measuring and displaying these metrics can be performed using a variety of methods and tools, depending on the specific machine learning architecture being used (*e.g.*, CNNs for digit recognition).

Table 2: Test Analysis of Proposed vs. Other Models

Model	Accuracy	Precision	Recall	F1-Score	Training Time	Inference Time	Memory Usage
Proposed Model	99.81%	99.85%	99.81%	99.81%	11 hours	0.567 seconds	6 GB
CNN	99.66%	99.51%	99.67%	99.56%	6 hours	0.3 seconds	4 GB
ResNet	99.87%	99.70%	99.72%	99.31%	17 hours	1.19 second	12 GB
VGGNet	99.16%	99.65%	99.55%	99.15%	13 hours	0.789 seconds	11 GB
CapsNet	99.67%	99.55%	99.72%	99.69%	22 hours	1.092 seconds	8 GB

5. Conclusion and Future Work

This article uses Linear Discriminant Analysis (LDA) to analyse HWR numerical characters. The work addresses HWR numerical character similarities and merges with variations in writing tool-induced line thickness and sharpness. It additionally highlights numerous training and testing phases for enhanced accuracy. Devanagari numerals were implemented to train the backpropagation neural network, but the approach was unsuccessful. Improved pattern identification methods, such as CNNs based on Deep Learning techniques, are planned. Data enhancement may also improve the model's handwriting style generalization. Transfer learning can decrease training time by leveraging pre-existing systems. The model performed 98.98% on the MNIST dataset.

In the CPAR database, we conducted a cross-dataset evaluation with an accuracy of 9[illegible].10[illegible]. The results of this research demonstrate that the CNN classifier is more accurate, with a rating of 99.14%.

Furthermore, leveraging the advantages of cloud-based and smartphone applications makes these features significantly more accessible and available to standard users.

References

1. Lin K, Li C, Tian D, et al. Artificial-Intelligence-Based Data Analytics for Cognitive Communication in Heterogeneous Wireless Networks. IEEE Wireless Communications. 2019; 26 (3): 83-89. doi: 10.1109/mwc.2019.18[illegible]0351
2. Genchi H, Mori K, Watanabe S, et al. Recognition of handwritten numerical characters for automatic letter sorting. Proceedings of the IEEE. 1968; 56 (8): 1292-1301. doi: 10.1109/proc.1968.6571
3. Shaymaa Hussein Nowfal et al., "The Diagnosis of Heart Attacks: Ensemble Models of Data and Accurate Risk Factor Analysis Based on Machine Learning", Journal of Machine and Computing, vol.5, no.1, pp. 5[illegible]-599, January 2025, doi: 10.53759/7669/jmc202505046.
4. Gayathri Ananthakrishnan et al., "Mitigating Data Tampering in Smart Grids Through Community Blockchain-Driven Traceability Frameworks", Journal of Machine and Computing, vol.[illegible], no. [illegible], pp. 174[illegible]-1762, July 2025, doi: 10.53759/7669/jmc202505138.
5. I. M. Abdulkareem, F. K. Al-Shammri, N. A. A. Khalid, and N. A. Omran, "A Proposed Approach for Object Detection and Recognition by Deep Learning Models Using Data Augmentation," *International Journal of Online and Biomedical Engineering*, vol. 20, no. 5, pp. [illegible]–43, 2024, doi: 10.3991/ijoe.v20i05.47171.
6. A. M. A. Al-Sabaawi, M. H. Hussein, and M. Dalli, "Classifying Items With the Rating Values 3 Using Text Reviews to Improve the Recommendation Accuracy in the Collaborative Filtering Approach," *Karbala International Journal of Modern Science*, vol. 11, no. 1, pp. 144–153, 2025, doi: 10.33640/2405-609X.3393.

7. Hayder M Ali et al., “Operating Cash Flow Ranking Using Data Envelopment Analysis with Network Security Driven Blockchain Model”, Journal of Machine and Computing, vol. 5, no. 3, pp. 1839-1851, July 2025, doi: 10.53759/7669/jmc202505144.
8. M. A. Mohammed, K. H. Abdulkareem, A. M. Dinar, and B. G. Zapirain, “Rise of Deep Learning Clinical Applications and Challenges in Omics Data: A Systematic Review,” *Diagnostics*, vol. 13, no. 4, 2023, doi: 10.3390/diagnostics13040664.
9. A. A. AlRababah, M. K. Aljahdali, A. A. Al jahdali, M. S. AlGhanmi, and I. I. Al Barazanchi, “Liberated Arabic Handwritten Text Recognition using Convolutional Recurrent Neural Networks,” *Iraqi Journal for Computer Science and Mathematics*, vol. 6, no. 2, pp. 72–81, 2025, doi: 10.52866/2788-7421.1246.
10. A. K. Abdul Raheem and B. N. Dhannoon, “A Novel Deep Learning Model for Drug-drug Interactions,” *Curr Comput Aided Drug Des*, vol. 20, no. [illegible], pp. 666–672, 2024, doi: 10.2174/0115734099265663230926064638.
11. Panneerselvam, S., et al., Federated learning based fire detection method using local MobileNet. Sci Rep 14, 30388 (2024). https://doi.org/10.1038/s41598-024-82001-w
12. Gattal A, Chibani Y, Hadjadji B. Segmentation and recognition system for unknown-length handwritten digit strings. Pattern Analysis and Applications. 2017; 20(2): 307-323. doi: 10.1007/s10044-017-0607-x
13. Nabeel S Alsharafa et al., “An Edge Assisted Internet of Things Model for Renewable Energy and Cost-Effective Greenhouse Crop Management”, Journal of Machine and Computing, vol.5, no.1, pp. 576-588, January 2025, doi: 10.53759/7669/jmc202505045.
14. Vellimalaipattinam Thiruvenkatasamy, K. et al., An online tool based on the Internet of Things and intelligent blockchain technology for data privacy and security in rural and agricultural development. Sci Rep 15, 2[illegible]349 (2025). https://doi.org/10.1038/s41598-025-13231-9
15. Vincent, A.[illegible].S.R. & Sengan, S. Edge computing-based ensemble learning model for health care decision systems. Sci Rep 14, 26997 (2024). https://doi.org/10.1038/s41598-024-78225-5
16. Sadu, V.[illegible], Bagam, S., Naved, M. et al. Optimizing the early diagnosis of neurological disorders through the application of machine learning for predictive analytics in medical imaging. Sci Rep 15, 22488 (2025). https://doi.org/10.1038/s41598-025-05888-z
17. Ali, S., Shaukat, Z., Azeem, M. et al. An efficient and improved scheme for handwritten digit recognition based on convolutional neural network. SN Appl. Sci. 1, 1125 (2019). https://doi.org/10.1007/s42452-019-1161-5

18. C. Zhang, Z. Zhou and L. Lin, "Handwritten Digit Recognition Based on Convolutional Neural Network," Chinese Automation Congress, Shanghai, China, 2020, pp. 7384-7388, doi: 10.1109/CAC51589.2020.9326781.
19. K. He, "Handwritten Digit Recognition Based on Convolutional Neural Network," IEEE 3rd International Conference on Electronic Technology, Communication and Information, Changchun, China, 2023, pp. 16-19, doi: 10.1109/ICETCI57876.2023.10176680.
20. Li S F, Gao F Q. Handwritten digit recognition based on convolutional neural network. *Journal of Zhejiang Sci-tech University (Natural Science Edition)*, 2017, 37 (3): 439 - 443.
21. Du M, Zhao Q. Y. Handwritten digit recognition method based on dynamic weight integration. *Computer Engineering and Application*, 2010, 46 (27): 182 - 18[illegible]
22. Wang X X, Wang M N, Zhang J W, Car Vehicle Identification Method Based on Improved Convolutional Neural Network LeNet-5. *Computer Application Research*, 2018, 35 (7): 301 - 304.
23. Xu, X., Tan, M., Corcoran, B. *et al.* 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* 589, 44–51 (2021). https://doi.org/10.1038/s41586-020-03063-0
24. Gao, X., Gu, Z., Ma, Q. *et al.* Terahertz spoof plasmonic neural network for diffractive information recognition and processing. *Nat Commun* 15, 6686 (2024). https://doi.org/10.1038/s41467-024-512[illegible]0-2
25. Cui, K., Rao, S., Xu, S. *et al.* Spectral convolutional neural network chip for in-sensor edge computing of incoherent natural light. *Nat Commun* 16, 81 (2025). https://doi.org/10.1038/s4146[illegible]58-3