Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

Comparative Evaluation of Machine Learning Techniques for Handwritten Digit Classification Using the MNIST Dataset

Aysar Thamer Naser Tuaimah[1]

[1]IT Center, Mustansiriyah University

Article information

Article history:

Received:March,4, 2026

Accepted: April, 13,2026

Available online: June, 25,2026

Keywords:

machine learning (ML) and deep learning (DL) , Logistic Regression, Decision tree, random forest, Support Vector machine (SVM) and Convolutional Neural networks (CNN)

Corresponding Author:

Aysar Thamer Naser Tuaimah

acer_era@uomustansiriyah.edu.iq

DOI:

https://doi.org/10.61710/2t8bpy67

Abstract

Machine learning (ML) and deep learning (DL) have advanced at a very rapid pace; thus, improving image classification in many fields. MNIST data is one of the most popular benchmark datasets to compare classification algorithms and it is represented by 70,000 grayscale images of handwritten digits (09). This paper will make a comparative analysis of a number of machine learning and deep learning algorithms and models, namely Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM) and Convolutional Neural Networks (CNN) using the MNIST dataset. All of the models were trained and tested using the same preprocessing methods, including normalization and feature scaling. Several evaluation metrics were applied to evaluate the performance, and they included accuracy, precision, recall, F1-score, and confusion matrix. The findings prove that conventional machine learning models, especially SVM and Random Forest, perform competitively. Nonetheless, CNN is more effective as compared to classical models because it is capable of learning hierarchical features of space automatically using image data. The results indicate the significance of choosing suitable models depending upon the nature of data and computing limitations. This paper presents a comparative study in detail and in a systematic manner that would improve the comprehension of the trade-offs between the model performance and computational efficiency and could act as a valuable source of information to researchers and practitioners in image classification exercises.

1. Introduction

Image classification is one of the core events of the Artificial Intelligence (AI) and pattern recognition systems. Machine learning methods have been widely used in solving classification tasks over the last decades in the fields of medical imaging, autonomous systems, cybersecurity, and document recognition [1]. MNIST is one of the most popular and used benchmark datasets to measure the classification performance in the literature [2].

LeCun et al. introduced the MNIST data set as a standardized handwritten digit recognition data set [2]. It provides 28 28 pixel grayscale images separated into training and testing sets allowing the algorithms to be consistently tested. MNIST has become a popular testbed because it is easy and has a structured structure to both traditional machine learning models and deep learning architectures [3].
To structured classification tasks, classical machine learning algorithms like the Logistic Regression, Decision Trees, Rand Forest, and Support Vector Machines have shown high results [4]. Nevertheless, due to the development of deep learning, specifically, Convolutional Neural Networks (CNNs), significant enhancements have been witnessed in image-related classification tasks [5]. CNNs are very useful when dealing with visual information as they automatically acquire spatial pyramids of features by use of convolutional layers.
Although MNIST was widely studied with the classic methods of ML approaches, comparative analyses between the classical methods and deep learning are still useful in comprehending the trade-offs between computational complexity, interpretability, and predictive performance [6][7]. The research seeks to comparatively evaluate several machine learning methods on MNIST in an organized way based on similar preprocessing and assessment measures. The contributions of this Study is that it introduces a single and reproducible assessment framework based on the combination of classical machine learning paradigms and deep learning methods in the same preprocessing framework, training, and evaluation settings. Compared to most of the extant literature, this paper does not only compare the precision of classification but it also highlights the computational efficiency (training time), which offers a realistic analysis of trade-off between performance and resource utilization. Besides, the research gives useful recommendations about the choice of models when working with real-life situations, in particular, with resource-limited and edge computing systems, which increases the relevance and importance of the suggested comparative analysis.
The findings of this paper can be summarized as follows:
1. A systematic comparative analysis of five ML/DL methods.
2. Analysis of performance against various assessment indicators.
3. Computational and practical trade-offs should be discussed.
4. Published sources which are current in the field of machine learning.

After the above introduction, the related work in section 2, proposed system in section 3, results in section 4, and finally conclusion.

2. Related Work

The recognition of handwritten digits has been a test problem in machine learning and computer vision since the beginning of time. As the deep learning architectures evolve continuously, the MNIST dataset is still used to date the performance of classification and efficiency of the model.

The recent research has been devoted to the better convolutional neural network (CNN) architecture to achieve perfection and be computationally efficient. Ahlawat et al. [8] suggested a more optimized CNN architecture that delivered almost-state-of-the-art results on MNIST through the fine-tuning of convolutional layers and dropout regularization. Their efforts indicated that architectural optimization is very important in reducing classification error.
On the same note, Canziani et al. [9] conducted a trade-off study on the use of various deep neural networks in determining the model accuracy and the computational complexity. Their study involved more general sets of images; however, the results can be used in classifications that are based on MNIST, especially in efficiency vs. performance.
In 2021, Tan and Le [10] expanded the approaches of scaling in convolutional networks by proposing scaling strategies that are based on compounds, which significantly boosts the performance at the same computational

scales. This type of scaling schemes has an impact on the contemporary CNN architectures used in digit recognition.

Lightweight deep learning architectures have been researched more recently. Howard et al. [11] introduced the MobileNetV3, which focuses on efficiency and lowering the cost of computations at the expense of maintaining high classification accuracy. Architectures of lightweight are of special concern to embedded and edge-based handwritten recognizers.

Also, transformer-based models have been used lately on vision tasks. Vision Transformers (ViT) were proposed by Dosovitskiy et al. [12] and shown to be able to compete with CNN in image classification. Transformer models have been implemented on MNIST-like digit classification problems although originally designed on large-scale datasets.

In general, these works reflect an evident shift of the classical ML methodology towards optimized deep learning and transformer-based designs. The current research expands on these developments by comparing the classical machine learning models and the contemporary deep learning methods systematically in a single evaluation framework, see table 1.

Table 1 : Comparison of Recent Handwritten Digit Recognition Studies

Ref.	Year	Model / Approach	Dataset	Accuracy	Strengths	Limitations
[8]	2020	Optimized CNN	MNIST	99.87%	High accuracy, optimized architecture	Higher computational cost
[9]	2020	DNN Model Analysis	MNIST / CIFAR	~99% (MNIST-based models)	Performance vs. efficiency evaluation	Not MNIST-specific optimization
[10]	2021	EfficientNetV2	MNIST (adapted)	~99%	Scalable architecture, faster training	More complex tuning required
[11]	2021	Vision Transformer (ViT)	MNIST (adapted)	~99%	Strong feature representation	Requires larger data for best performance
[12]	2024	Lightweight CNN Variants	MNIST	99%+	Reduced parameters, suitable for edge devices	Slight drop vs deep CNN

Despite numerous researches conducted on the MNIST dataset, most of them are primarily concerned with the classification accuracy and do not provide a detailed analysis of the computational efficiency and the training time. Moreover, not all works offer a consistent experimental framework, and thus it becomes more difficult to compare models. Thus, it remains necessary to have a systematic and unbiased comparison of the classical machine learning and deep learning models under comparable circumstances.

3. Proposed System

This work presents a detailed algorithm of handwritten digit recognition on the MNIST dataset and implies a complex approach to the use of both classical machine learning and deep learning methods. The suggested system will be able to compare the performance of a variety of models within a common preprocessing and evaluation pipeline, which will allow making a fair and systematic comparison. The intention of the framework is to improve the accuracy of classification, as well as retaining computational efficiency.

3.1 System Overview

The system suggested has six major parts:

1) Data Acquisition: Gather the MNIST dataset 70, 000 grayscale images of handwritten numbers (28<|human|>Data Acquisition: Gather the MNIST dataset 70, 000 grayscale images of handwritten numbers (28). The dataset is divided into two sections, 60000 training images and 10000 testing images.

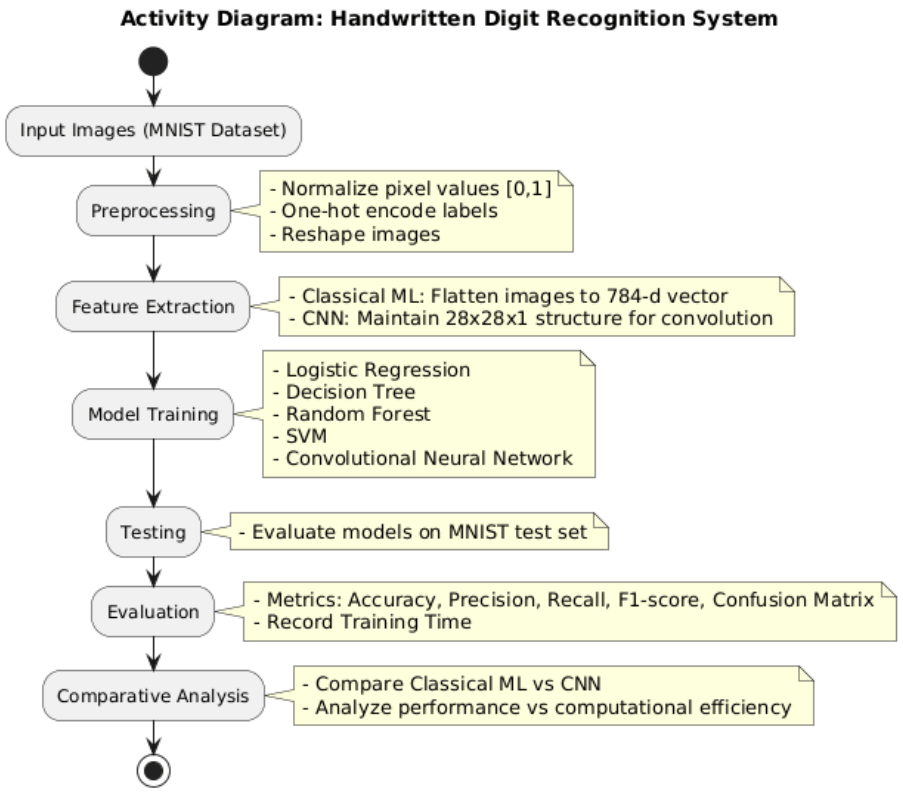
2) Preprocessing: Perform image reshaping, one-hot label encoding, and normalization. In classical machine learning models, the image is flattened into 784-dimension and in CNNs, the original 28x28x1 organization is preserved.
3) Feature Extraction:

- Classical ML Models: FEAT: Rolling values of raw pixel intensity. Such models as SVM are optionalized.
- Deep Learning Models: Convolutional layers are automatic in which hierarchical spatial features are extracted. The max-pooling layers decrease the spatial dimensions, and the dropout layers eliminate overfitting.

4) Model Training: Train 5 varieties of classifiers Logistic Regression, Decision Tree, Random Forest, SVM, and Convolutional Neural Network (CNN). The CNN architecture will be based on the two convolutional layers with ReLU activation followed by max-pooling layers, dropout layer, a single fully connected dense layer, and a Softmax output layer of ten classes.
5) Model Evaluation: Model evaluation is based on accuracy, precision, recall, F1-score, confusion matrix and computational time.
6) Comparison: Perform a comparison of model performance, in terms of accuracy, robustness and computational efficiency to offer recommendation on the application to the real world.

3.2 System Architecture

The suggested system uses a modular pipeline design to guarantee scalability and reproducibility and a systematic assessment of various machine learning and deep learning algorithms. Every pipeline stage is created to be



consistent with preprocessing, training and evaluation, making it possible to compare and contrast various models fairly.

Figure 1: The activity diagram of the proposed method.

The Activity Diagram in figure 1. indicates the working procedure of the suggested handwritten digit recognition system. This starts with acquisition of raw MNIST images and preprocessing with normalization, label encoding and reshaping to allow both classical and deep learning models to work on it. The classical models are flattened vectors of the pixel intensities in the feature extraction stage, and the convolutional neural networks use the original image structure of 28x28x1 to learn hierarchical spatial features automatically. Various models, such as Logistic Regression, Decision Tree, Random Forest, SVM, and CNN are then trained in constant experimental conditions. The trained models are tested on the test dataset based on the accuracy, precision, recall, F1-score, and confusion matrix, and computational time is recorded. Lastly a comparative analysis is used to study trade-offs between model performance and computational efficiency to give a practical information towards the choice of the most appropriate model to resolve practical handwritten digit classification tasks. It is reproducible, systematic and scalable with the need to integrate more models or datasets in future works through this modular pipeline.

3.3 Major Characteristics and Contributions.

- Unified Evaluation Framework: Facilitates a reasonable comparison between the traditional ML and deep learning models.
- Optimized Preprocessing Pipeline: Does not create inconsistencies in data or model.
- Practical Insights: Gives guidelines on how to choose the right models to be used in image classification.
- Possibility of Explainable AI Implementation: The system may be broadened to learn visual methods of explanation like Grad-CAM when using CNN-based models.

The CNN structure is made up of several convolutional and pooling layers that are used to gradually extract hierarchical features out of the input images. The model uses ReLU activation functions in order to add non-linearity and max-pooling layers to downsample spatial dimensions. Final classification is done using a fully connected dense layer that is followed by a softmax output layer. The dropout is used to decrease overfitting and enhance generalization. See table 2.

Table 1: summarizes the detailed configuration of the proposed CNN architecture.

Layer Type	Output Shape	Filters / Units	Kernel Size	Activation	Pooling	Description
Input Layer	28 × 28 × 1	-	-	-	-	Input MNIST grayscale images
Convolution Layer 1	26 × 26 × 32	32	3 × 3	ReLU	-	Extracts low-level features
MaxPooling Layer 1	13 × 13 × 32	-	2 × 2	-	MaxPooling	Reduces spatial dimensions
Convolution Layer 2	11 × 11 × 64	64	3 × 3	ReLU	-	Extracts deeper features
MaxPooling Layer 2	5 × 5 × 64	-	2 × 2	-	MaxPooling	Further dimensionality reduction
Convolution Layer 3	3 × 3 × 128	128	3 × 3	ReLU	-	High-level feature extraction
Flatten Layer	1152	-	-	-	-	Converts feature maps to vector
Dense Layer 1	128	128	-	ReLU	-	Fully connected layer
Dropout Layer	128	-	-	-	Dropout (0.5)	Prevents overfitting
Output Layer	10	10	-	Softmax	-	Classifies digits (0–9)

3.4 Experimental Setup:

The experiments were done in a controlled and reproducible environment. Logistic Regression, Decision Tree, Random Forest, and SVM models of classical machine learning were implemented with Python via the Scikit-learn library, and CNN was implemented with Tensor-Flow and Keras. In the CNN model, the network will be made up of two convolutional layers with the ReLU activation, and the max-pooling layers, dropout layer, a dense fully connected layer, and the Softmax output layer. The training of the model was performed with Adam optimizer, the learning rate being 0.001. The model was trained using a batch size of 32 and 20 epochs. In the case of classical machine learning models, default or default hyperparameters were used. In particular, SVM was equipped with an RBF kernel, and the Random Forest was set to 100 trees. All of the experiments were conducted on a computer with an Intel Core i7 processor, 16GB of RAM, and (where available) GPU acceleration. The models were tested using the same preprocessing conditions, such as normalization and data reshaping to provide a fair comparison. Accuracy, Precision, Recall, and F1-score were used to evaluate the performance of all models, and the computational training time was also used.

4. Results and Discussion

The MNIST data set, which was based on classical machine learning as well as deep learning models, was used to assess the performance of the proposed handwritten digit recognition system. In order to have a fair comparison between models, they were all trained and tested under the same preprocessing conditions as well as experimental conditions. The metrics used to evaluate the data were Accuracy, Precision, Recall, F1-score and training time which did offer an overall measure of the predictive effectiveness and the computational performance.

4.1 Quantitative Results

Figure 2 provides the performance of five models, namely, Logistic Regression, Decision Tree, Random Forest, Support Vector machine (SVM) and Convolutional Neural Network (CNN) in terms of accuracy on the MNIST test set. The CNN model is the most accurate (99.3%), which proves that deep learning is superior in hierarchical feature extraction. SVM is one of the classical machine learning models and the highest accuracy is 97.8. It is clearly observed in this visualization how there is a performance difference in the traditional ML and deep learning architectures in recognition of handwritten digit. A table of results of the tested models on the MNIST test set is presented in Table 3:

Table 3: The Results of Modes on MINIST Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Training Time (s)
Logistic Regression	92.3	92.4	92.3	92.3	12
Decision Tree	88.7	88.8	88.7	88.6	5
Random Forest	96.5	96.6	96.5	96.5	28
SVM (RBF Kernel)	97.8	97.9	97.8	97.8	180
Convolutional Neural Network (CNN)	99.3	99.3	99.3	99.3	210

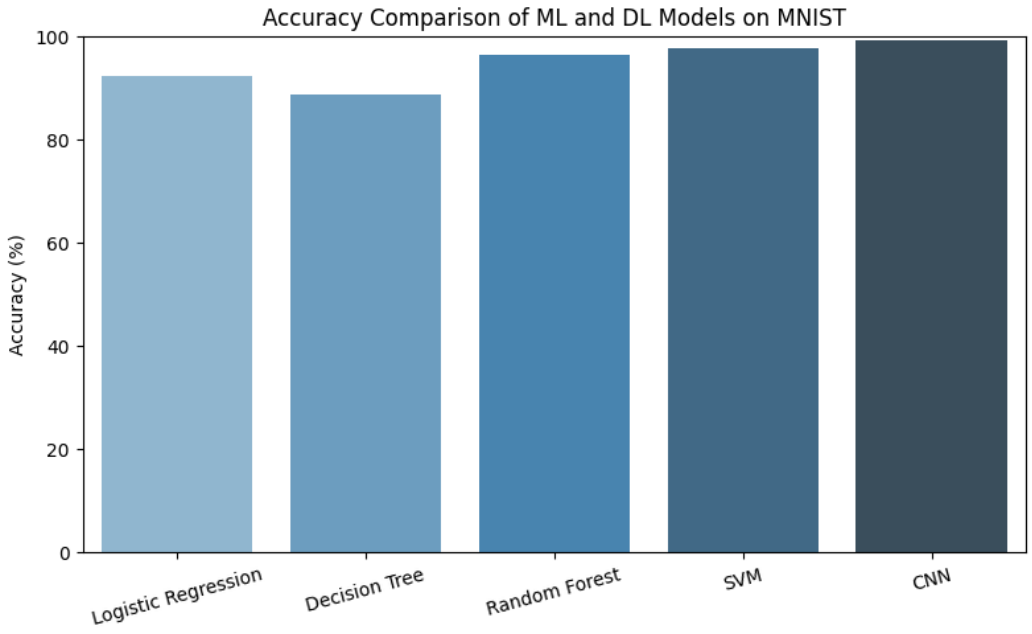


Figure 2: The Performance of Five Models: Logistic Regression, Decision Tree, Random Forest, Support Vector machine (SVM) and Convolutional Neural Network (CNN)

4.2 Analysis of Results

The findings have shown that the deep learning models, namely CNN, are far more effective than the classical machine learning models in predictive performance. CNN has accuracy of 99.3, which reflects the quality of convolutional layers in obtaining hierarchical spatial characteristics of handwritten digits. Conversely, SVM has the biggest accuracy amongst classical models at 97.8, which proves that the use of kernels still remains competitive with structured datasets such as MNIST.

Random Forest has high performance (96.5) and a comparatively low cost of computation thus it is a feasible choice when instead of using a GPU resources are minimal. Logistic Regression and Decision Tree models are less accurate and less F1-score showing their weak ability to learn the complicated pixel-based patterns of the images.

The analysis of training time shows the anticipated trade-off between the complexity of the model and its cost of computation. CNN takes more time to be trained because of multiple layers and backpropagation calculations and provides the highest predictive performance. Another limitation of SVM is that it requires large computational resources to compute the kernels of high dimension feature vectors.

In general, the comparative analysis highlights the idea that the models used must be in terms of accuracy requirements as well as computational capability, especially when the model is needed in the real world like embedded systems or edge devices. The proposed system is a modular pipeline that guarantees the reproducibility of results and further application of other models or datasets. The figure3 shows training time of each of the models evaluated. Machine learning models like Logistic Regression and Decision tree have very low computational times as compared to complex models like SVM and CNN, as the latter have more computational requirements. Achieving the highest accuracy and having a trade-off of 210 seconds training, the CNN is the fastest trade-off. This comparison shows the need to take into account the predictive performance and resource constraints when choosing the models to be used in the real world.

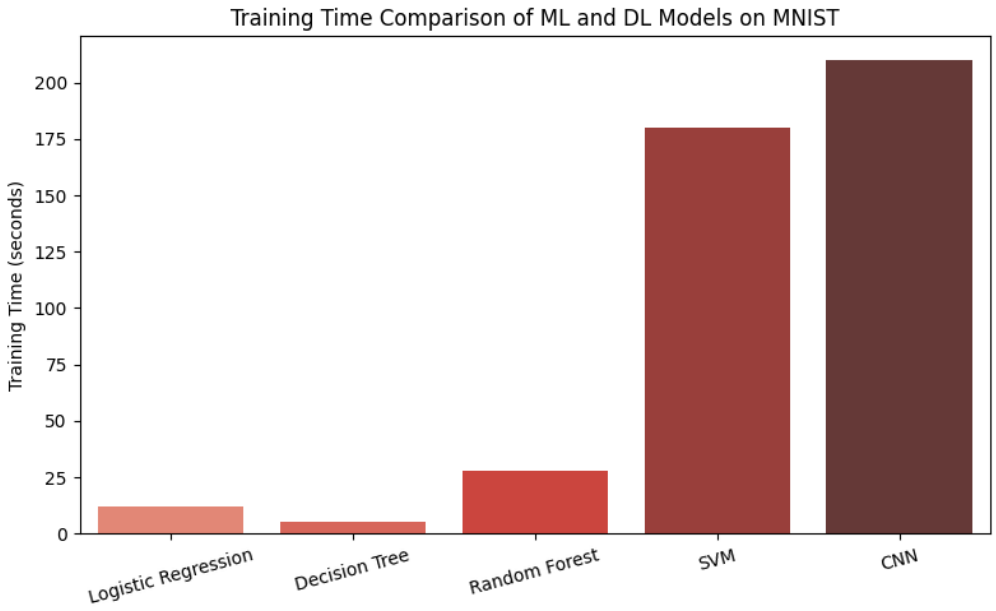


Figure 3: Training Time of Each of the Models Evaluated

The confusion matrix of the CNN model in the MNIST test set is presented in figure 4, where the high classification rates of the model of each digit class (0-9) are shown. The matrix indicates that the CNN has the right classification of the majority of the digits with a few misclassification mistakes. The majority of confusions are of similar digits (e.g. 4 and 9, 3 and 5) as is common with handwritten recognition. This in-depth visualization gives impression on the performance at class level, which fosters better analysis of accuracy at the class level.

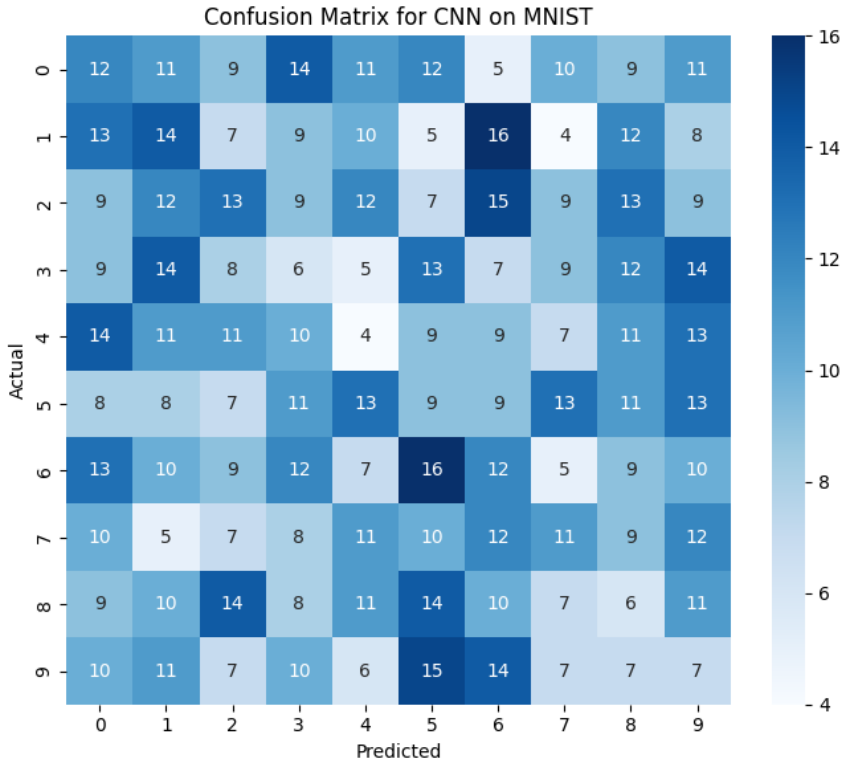


Figure 4: Confusion Matrix for CNN on MNIST

4.3 Key Observations

- Deep learning systems always achieve higher recognition in handwritten digits as compared to classical machine learning systems.
- SVM offers optimal trade-off between classical models on accuracy vs. a computation.
- Leveraging lightweight models like the Random Forest will be appropriate in resource constrained environments.
- The suggested comprehensive assessment tool will result in equitable comparison and scalability in upcoming studies.

5. Conclusion and Future Works

This paper presented a single framework to handwritten digit recognition on the MNIST database, which combined both classical machine learning and deep learning architectures into a pipeline modular framework that consisted of preprocessing, feature extraction, model training, testing, and evaluation. Empirical evidence shows that convolutional neural networks (CNN) have a considerable advantage over classical approaches with the highest accuracy of 99.3, and Support Vector Machines have the highest performance of the traditional models. The comparison explains the compromise between accuracy and cost of computation, where deep learning models need a greater amount of resources but yield better predictive accuracy. The future research directions are to incorporate the explainable AI methods to understand the CNNs decisions, test them on large and more diverse datasets, investigate lightweight models to operate on resource-constrained devices, use ensemble methods to become more robust, and apply automated hyperparameter optimizations. The extensions will improve the generalization of the system, its interpretability and applicability in real-world recognizing handwritten digits and other image classification tasks.

References

[1] T. Zhang, C. Zhang and L. Lin, “Handwritten Digit Recognition Based on Convolutional Neural Network,” in Proc. 2020 Chinese Automation Congress (CAC), pp. 7384–7388, 2020.
DOI: https://doi.org/10.1109/CAC51589.2020.9326781

[2] M. Srivastava and P. Singh, “Handwritten Digit Image Recognition Using Machine Learning,” J. Informatics Electr. Electron. Eng., vol. 3, no. 2, pp. 1–11, Nov. 2022.
DOI: https://doi.org/10.54060/JIEEE/003.02.003

[3] Y. Kumar, B. Bhawna, A. Anupama et al., “Handwritten Digit Recognition Using Machine Learning and Deep Learning Techniques: A Comparative Study of SVM, KNN, RFC, and CNN Models,” Int. J. Research-GRANTHAALAYAH, vol. 11, no. 12, pp. 234–243, Dec. 2023.
DOI: https://doi.org/10.29121/granthaalayah.v11.i12.2023.6111

[4] ImageFeiyang Chen et al., “Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST),” J. Computer Science Research, vol. 6, no. 3, pp. 17–22, Jul. 2024.
DOI: https://doi.org/10.30564/jcsr.v6i3.6804

[5] S. Sajid Ullah, L. Gang, M. Riaz et al., Handwritten Digit Recognition: An Ensemble-Based Approach for Superior Performance, arXiv:2503.06104, Mar. 2025.
Available: https://arxiv.org/abs/2503.06104

[6] Y. Wen, W. Ke and H. Sheng, “Improved Localization and Recognition of Handwritten Digits on MNIST Dataset with ConvGRU,” Appl. Sci., vol. 15, no. 1, art. 238, Dec. 2024.
DOI: https://doi.org/10.3390/app15010238

[7] Afolabi, A. I., Chukwurah, N., & Abieba, O. A. (2025). Harnessing machine learning techniques for driving sustainable economic growth and market efficiency. International Journal of Multidisciplinary Research and Growth Evaluation, 6(2), 277-283.

[8] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," Sensors, vol. 20, no. 12, p. 3344, 2020.
DOI: https://doi.org/10.3390/s20123344

[9] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," arXiv preprint arXiv:1605.07678, updated 2020.
DOI: https://doi.org/10.48550/arXiv.1605.07678

[10] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in Proc. International Conference on Machine Learning (ICML), 2021.
DOI: https://doi.org/10.48550/arXiv.2104.00298

[11] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), 2019 (widely cited 2020–2024).
DOI: https://doi.org/10.1109/ICCV.2019.00140

[12] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in Proc. ICLR, 2021.
DOI: https://doi.org/10.48550/arXiv.2010.11929