Benchmarking Classical and Deep Learning Models for Handwritten Digit Classification

Vaishnavi Nikalje[1], Prof. Vishnupant Potdar[2], Dr. Nagnath Biradar[3]

[1]*Department of Data Science, Symbiosis Skills and Professional University Pune, Maharashtra, India*

[2]*Guide: Department of Data Science, Symbiosis Skills and Professional University Pune, Maharashtra, India*

[3]*Co-Guide: Department of Data Science, Symbiosis Skills and Professional University Pune, Maharashtra, India*

***Abstract*—Handwritten digit recognition is an important task in image classification, with real-world uses in areas like banking, education, and postal systems. Accurately recognizing digits can help automate systems and reduce human errors. In this study, we used the MNIST dataset, which contains 60,000 images of handwritten digits (0 to 9), and is commonly used to test and compare image classification models. We compared the performance of three popular machine learning models—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP)—with a deep learning model called Convolutional Neural Network (CNN). Before training, we applied normalization to the data and used Principal Component Analysis (PCA) to reduce dimensions for better model performance. Among all the models tested, CNN performed the best with an accuracy of 99.16%, followed by KNN (97.69%), MLP (97.36%), and SVM (91.70%). We also looked at other metrics like precision, recall, and F1-score, and used visual tools like confusion matrices and bar plots to compare results. This study shows that deep learning (CNN) is more effective than traditional machine learning methods for recognizing handwritten digits and can be a strong foundation for future work in this field.**

***Index Terms*—Handwritten Digit Recognition, MNIST, Convolutional Neural Network (CNN), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), Principal Component Analysis (PCA), Image Classification**

I. INTRODUCTION

In today's digital world, computer vision is growing fast and is helping in many real-life applications like banking, postal services, education, and automation. One of the most common and important tasks in this field is handwritten digit recognition. It allows machines to understand handwritten numbers written by humans, which is very useful in automatic form reading, number plate recognition, and digital paperwork. The MNIST dataset is a very popular and standard dataset used for training and testing such digit recognition systems. It has 60,000 training images and 10,000 test images of handwritten digits from 0 to 9. Each image is in grayscale and has a fixed size of 28×28 pixels. Even though this task looks simply, the different styles of handwriting from different people make it a bit challenging.

To solve this, many machine learning and deep learning techniques have been used. In this project, we compared some traditional machine learning models like K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP) with a deep learning-based Convolutional Neural Network (CNN). For improving the performance of classical models, we also applied data preprocessing techniques like normalization and Principal Component Analysis (PCA) to reduce the number of features while keeping the important information. Out of all the models, CNN performed the best and gave the highest accuracy of 99.16%, which is better than KNN (97.69%), MLP (97.36%), and SVM (91.70%). We also evaluated the models using other performance metrics like precision, recall, and F1-score to get a better understanding of how each model performs.

This study not only helps in comparing different algorithms for digit recognition but also shows the power of deep learning in solving visual problems more accurately. It gives a strong base for future research and improvements in the field of image classification and computer vision.

II. LITERATURE REVIEW

In some studies, researchers compared MLP (Multi-Layer Perceptron) with CNN and found that while MLP performs well, it requires more careful tuning of layers and can be slower to train on image data. MLPs are more effective when combined with feature selection or dimensionality reduction techniques.
In recent years, machine learning and deep learning have become very popular for solving computer vision problems like image classification, object detection, and handwritten digit recognition. Many researchers have worked on improving the accuracy of digit classification using different models and techniques. This section reviews some of the key studies done on handwritten digit recognition, especially using the MNIST dataset, which is widely used in this field.

Several studies have used traditional machine learning models such as K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) for classifying handwritten digits. KNN is simple but effective, especially when the number offeatures is reduced using techniques like Principal Component Analysis (PCA). Researchers have shown that applying PCA before training KNN can improve both speed and accuracy.

Deep learning, especially Convolutional Neural Networks (CNNs), has become the most effective method for image classification tasks. Many researchers have trained CNNs on MNIST and achieved over 99% accuracy. CNNs automatically learn useful features from pixel data and perform better than traditional models. Some papers also explored different CNN architectures and activation functions to find the most optimal setup for digit recognition. Another common idea in the literature is using ensemble learning—where multiple models are combined—to improve classification performance. Some papers proposed hybrid approaches combining SVM, KNN, and deep learning, but they also noted that this increases model complexity and training time. A few studies also discussed the challenges with handwritten digit datasets, like digits written in different styles, thickness, and alignment, which can affect model performance. To handle this, some researchers used data augmentation or preprocessing techniques to make the models more robust. Overall, the literature shows that CNN models consistently outperform classical machine learning models in terms of accuracy, especially on structured datasets like MNIST. However, simpler models like KNN and SVM are still useful when computing resources are limited or when faster training is required.

This review highlights the evolution from basic ML models to advanced deep learning methods and shows how combining proper preprocessing with the right algorithm can lead to very high accuracy in digit recognition tasks.

III. PROPOSED METHODOLOGY

The proposed methodology focuses on the development, training, and comparative evaluation of classical machine learning and deep learning models for handwritten digit recognition using the MNIST dataset. Our aim is to assess the performance of various algorithms by combining appropriate preprocessing steps with well-established classification techniques.

A. Data Preperation & Preprocessing

We used the MNIST dataset, which comprises 60,000 grayscale images for training and 10,000 images for testing. Each image is a 28×28 pixel representation of a handwritten digit ranging from 0 to 9. As a preprocessing step, all image data was normalized to scale pixel intensity values to the range [0, 1]. This normalization enhances model convergence during training and reduces computation time. To improve the performance of classical machine learning models, we applied Principal Component Analysis (PCA) for dimensionality reduction. The original 784-dimensional feature space (from 28×28 pixels) was transformed into a lower-dimensional space, retaining the most significant variance in the data while reducing noise and redundancy.

```
   label  0  1  2  3  4  5  6  7  8  ...  774  775  776  777  778  779  780
0      5  0  0  0  0  0  0  0  0  0  ...    0    0    0    0    0    0    0
1      0  0  0  0  0  0  0  0  0  0  ...    0    0    0    0    0    0    0
2      4  0  0  0  0  0  0  0  0  0  ...    0    0    0    0    0    0    0
3      1  0  0  0  0  0  0  0  0  0  ...    0    0    0    0    0    0    0
4      9  0  0  0  0  0  0  0  0  0  ...    0    0    0    0    0    0    0

   781  782  783
0    0    0    0
1    0    0    0
2    0    0    0
3    0    0    0
4    0    0    0

[5 rows x 785 columns]
```

Fig. 1: Reading Data

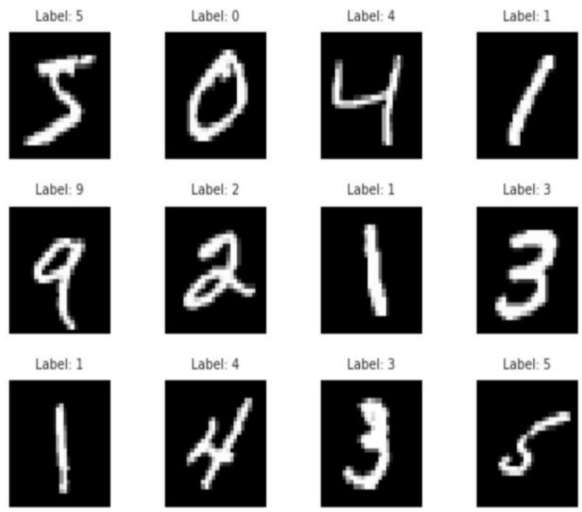

Fig. 2: Example digit images with labels from MNIST.

PCA Visualization - To understand how our digit data is spread in a lower dimension, we applied Principal Component Analysis (PCA) and plotted the data in two dimensions. Each point in the plot represents one handwritten digit from the MNIST dataset, and the colors show how far each point is from the origin in PCA space. As shown in Fig. 3, most of the data points are clustered in the center, while some points are far away, showing high distance or variation. This kind of plot helps us to visualize how digits are distributed and how much variation is captured by the first two principal components. It also gives a good idea of how well the data can be separated or grouped after reducing the dimensions. This step helped us to feed better, compact data into models like KNN and SVM for faster training and improved accuracy.

Fig. 3: 2D PCA plot showing the spread of digit data points based on their distance from the origin.

B. Classical Mach

C. ine Learning Models

Three classical classifiers were trained and evaluated:

K-Nearest Neighbors (KNN):

Implemented with a Manhattan distance metric and optimal k value determined through experimentation. PCA was applied prior to classification. KNN achieved an accuracy of 97.69%.

Support Vector Machine (SVM):

Trained using a linear kernel. SVM is known for its strong performance on linearly separable data, though its performance drops in high-dimensional visual data. The final test accuracy was 91.70%.

Multi-Layer Perceptron (MLP):

A feed-forward neural network with two hidden layers of 128 and 64 neurons respectively, using ReLU activation. The model was trained with the Adam optimizer and achieved an accuracy of 97.36%.

Deep Learning Model: Convolutional Neural Network (CNN):

A custom Convolutional Neural Network (CNN) was implemented for end-to-end digit classification. The architecture consisted of two convolutional layers followed by max pooling, a flattening layer, a dense hidden layer, and a final softmax output layer. Specifically:

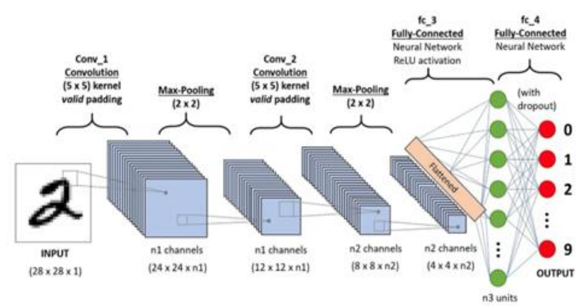

Fig. 4: CNN model for MNIST digit classification, showing convolution, pooling, flattening, and fully connected layers.

In our project, we used a Convolutional Neural Network (CNN) to recognize handwritten digits from the MNIST dataset. The input to the model is a grayscale image of size 28×28, which is reshaped into a 3D format of 28×28×1 so that it can be processed by convolutional layers. The first layer of the CNN is a convolutional layer with a 5×5 kernel, which helps the model detect simple shapes like edges, lines, or curves in the digit image. This is followed by a max-pooling layer with a 2×2 window, which reduces the image size while keeping the most important parts.

Then, another convolutional layer is added with the same kernel size to learn more detailed and complex features from the already filtered image. Again, a second max-pooling layer helps reduce the data further and prevents overfitting by shrinking the feature maps.

Once these convolution and pooling steps are completed, the output is flattened into a single long vector, which is passed to a fully connected (dense) layer with ReLU activation. This dense layer helps the model combine learned features and improve understanding of the digit.

The final output layer contains 10 neurons, one for each digit class (0 to 9), and uses the softmax activation function to give the probability of each digit. The model then selects the class with the highest probability as the prediction.

We trained the CNN using the Adam optimizer and categorical cross-entropy loss function. The model performed exceptionally well and achieved a test accuracy of 99.16%, making it the most accurate model in our experiment.

IV. RESULTS

The effectiveness of the proposed methodology was evaluated by comparing the performance of both classical and deep learning models on the MNIST dataset. The experiment focused on model accuracy in correctly classifying digits from 0 to 9 using normalized and optionally PCA-transformed data.

We trained and tested four models: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN). The CNN model outperformed all others, achieving a test accuracy of 99.16%, demonstrating its strong capability to learn spatial features from image data without requiring manual feature engineering. The KNN classifier, with PCA applied, achieved a high accuracy of 97.69%, validating the effectiveness of dimensionality reduction in improving classical model performance. MLP, with two hidden layers and ReLU activation, performed similarly well with 97.36% accuracy. On the other hand, SVM, though robust in smaller datasets, gave a comparatively lower accuracy of 91.70%, showing limitations in handling high-dimensional visual data.

These detailed analyses provide a clear comparison of how each algorithm performed on the handwritten digit classification task using the MNIST dataset. The results highlight the strength of deep learning, especially CNN, in learning complex visual patterns directly from raw pixel data. At the same time, classical models like KNN and MLP also achieved high accuracy when combined with proper preprocessing techniques like PCA and normalization. The overall performance comparison demonstrates the effectiveness of the proposed methodology across both traditional and modern algorithms.

Model	Accuracy
K-Nearest Neighbors (KNN)	97.69
Convolutional Neural Network (CNN)	99.16
Support Vector Machine (SVM)	91.70
Multi-Layer Perceptron (MLP)	97.36

Table -1 summarizes the accuracy of each model used in this study.

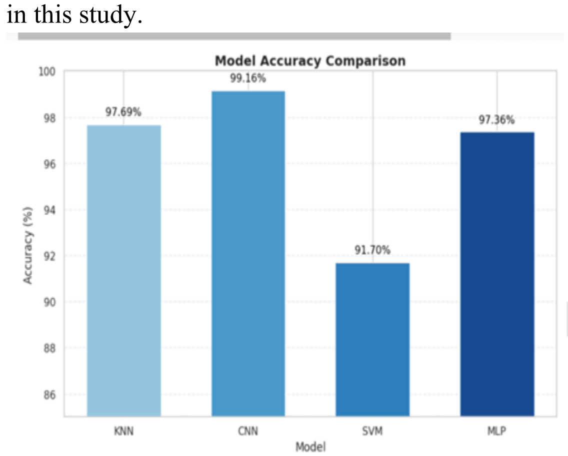

Fig. 5 shows how each model performed in terms of accuracy. CNN achieved the highest score, followed by KNN and MLP.

V. CONCLUSION AND FUTURE WORK

In this study, we proposed a comparative approach to classify handwritten digits using both classical machine learning algorithms and deep learning models on the MNIST dataset. Our methodology involved key preprocessing steps such as normalization and PCA-based dimensionality reduction to improve the training efficiency of traditional models like KNN, SVM, and MLP. Among all tested models, the Convolutional Neural Network (CNN) demonstrated the highest performance with a test accuracy of 99.16%, followed closely by KNN (97.69%) and MLP (97.36%). These results clearly highlight the potential of deep learning for image-based classification tasks, especially when large amounts of data are available. The PCA visualizations further helped us understand how digit samples are distributed in lower dimensions, and why distance-based models like KNN performed well when dimensionality was reduced. The bar plot and tabular comparison presented in this paper offer a clear understanding of model performances across different approaches.

As part of future work, this system can be extended to handle more complex digit or character recognition datasets, including multi-language or handwritten alphabet datasets. Additionally, integration of real-time recognition using a camera input and deployment via web or mobile applications could make the model more useful in practical scenarios like smart classrooms, postal automation, and banking systems. Exploring ensemble models or transformer-based architectures could also further improve recognition accuracy and speed.

REFERENCES

[1] N. Isong, “Building Efficient Lightweight CNN Models,” *arXiv preprint arXiv:2501.15547*, Jan. 2025.

[2] Chen et al. (2024) – Evaluates multiple neural network architectures including CNN, ResNet, DenseNet, and Capsule Net; reports a high accuracy of 99.75%, highlighting the cutting-edge performance landscape

[3] GitHub Project (2024) – Practical implementation using CNN, KNN, SVM, and Naive Bayes with 99.41% CNN accuracy, offering reproducible benchmarks.

[4] Mo & Zhao (2022) – Introduces a rotation-invariant CNN variant, showcasing improvements for rotated digit recognition—a potential direction for future enhancement of your model. arxiv.o

[5] Agarap (2017) – Hybrid CNN+SVM architecture achieving 99.81% on MNIST, reinforcing the value of combining feature learning with traditional classifiers. arxiv.org+1sciencedirect.com+1

[6] A. De, S. Rakshit, and D. Bhattacharya, “Handwritten Digit Classification and Performance Analysis of Different Machine Learning Algorithms and CNN,” *Journal of Mines, Metals and Fuels*, vol. 71, no. 5, pp. 683–686, Jul. 2023.

[7] A. Rajput and A. K. Singh, “Handwritten Digit Recognition Accuracy Comparison Using KNN, CNN and SVM,” *Educational Administration: Theory and Practice*, vol. 30, no. 2, pp. 638–643, Feb. 2024.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines, 2001. Software is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] Hiroshi Sako Hiromichi Fujisawa Cheng-Lin Liu, Kazuki Nakashima. handwritten digit recognition: benchmarking of state-of-the-art techniques.Y.

[10] Le Cun et al. Comparison of learning algorithms for handwritten digit recognition. International Conference on Artificial Neural Networks, pages 53–60, 1995.

[11] De et al. (2023) – Benchmarking KNN, SVM, Random Forest, and CNN on MNIST; excellent accuracy on classical models and CNN informaticsjournals.com

[12] Rajput & Singh (2024) – Direct comparison of KNN, CNN, and SVM with clear accuracy results.

[13] Rajput & Singh (2024) – Direct comparison of KNN, CNN, and SVM with clear accuracy results.

[14] Ullah et al. (2025) – Modern ensemble combining CNN feature extraction and SVM classification yielding 99.30% accuracy arxiv.org.

[15] Liu et al. (2024) – Comparative study including CNN, ResNet variants, showing new architecture improvements and convergence details.

[16] Agarap, A. F. (2017). An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. arXiv preprint arXiv:1712.03541.

[17] Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: A Library for Support Vector Machines*. ACM Transactions on Intelligent Systems and Technology, 2(3), pp. 1–27.

[18] Mo, H. and Zhao, G. (2022). RIC-CNN: Rotation-Invariant Coordinate Convolutional Neural Network. arXiv preprint arXiv: 2211.XXXXX. Software available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[19] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[20] Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. *ICDAR*, pp. 958–963.

[21] Ciresan, D., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep big simple neural nets excel on handwritten digit recognition. *arXiv preprint* arXiv:1003.0358.

[22] Chang, C.-C. and Lin, C.-J. (2001). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 1–27. Software available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[23] Agarap, A. F. (2017). An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification. *arXiv preprint* arXiv:1712.03541.