International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 2, Mar-Apr 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Comparative Analysis of Machine Learning Models for Handwritten Digit Recognition

Reda Elgsir*, Abdelilah Ajhil**
*(School of Computer Science, Nanjing university of Information Science and Technology
Email: redaelgsir8@gmail.com)
** (School of Computer Science, Nanjing university of Information Science and Technology
Email: abdilah.ajhil190@gmail.com)

----------**********----------

Abstract:

Handwritten digit recognition plays a pivotal role in modern computer vision and its applications in optical character recognition systems. This paper provides an analysis of different machine learning models applied to classify handwritten digits using the MNIST dataset. The dataset consists of features extracted from grayscale images of digits (0-9). This study compares and evaluates the performance of various models using metrics such as accuracy and F1-score. Ensemble modelling techniques are also explored to enhance prediction accuracy and reliability. The findings underline the effectiveness of advanced models like convolutional neural networks in digit classification tasks and their potential in practical applications.

Keywords — **Handwritten Digit Recognition, Machine Learning, Computer Vision, OCR, CNN, MNIST Dataset, MNIST Dataset, Classification Models, Ensemble Learning.**

----------**********----------

I. INTRODUCTION

Handwritten digit recognition is a foundational problem in computer vision with wide-ranging applications, such as automated postal sorting, banking systems for check pro cessing, and digital data entry. The goal is to classify digit images into one of ten categories (0–9). Despite its apparent simplicity, the challenge arises from variations in handwriting styles, scales, rotations, and noise, which demand robust and efficient classification techniques. Historically, traditional approaches relied on manually engineered features and algorithms like k-Nearest Neighbours (k-NN) and Support Vector Machines (SVM)[4]. While effective for smaller datasets, these methods often struggled with scalability and generalization. The advent of deep learning, especially Convolutional Neural Networks (CNNs), revolutionized this field by enabling models to automatically learn feature hierarchies from raw data. CNNs have set new benchmarks in accuracy and efficiency, outperforming classical methods by significant margins. The MNIST dataset, a widely used benchmark for hand written digit recognition, consists of 60,000 training and 10,000 testing images of 28x28 grayscale digits. Despite its simplicity, MNIST remains an essential dataset for evaluating machine learning models. This study employs MNIST to analyse and compare traditional algorithms and modern CNN architectures. Preprocessing techniques like normalization, augmentation, and noise reduction are applied to enhance input data quality. Additionally, oversampling ad dresses class imbalance, and ensemble modelling is explored to improve overall performance. Performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. The study emphasizes the trade-offs between accuracy, computational efficiency, and scalability, offering insights into selecting suitable approaches for various applications. By examining the strengths and limitations of traditional and modern methods, this research highlights the transformative impact of deep learning while acknowledging the continued relevance of traditional approaches in specific

 ©IJSRED: All Rights are Reserved

scenarios. Handwritten digit recognition remains a critical benchmark for advancing machine learning, and this study contributes to the development of robust and adaptable solutions for real-world applications.

II. RELATED WORK

Handwritten digit recognition has been a focus of numerous studies, leveraging both traditional machine learning techniques and modern deep learning approaches. Early studies emphasized the use of traditional methods like Support Vector Machines (SVMs) and k-Nearest Neighbours (k-NN). These algorithms provided robust baselines, with k-NN offering simplicity and ease of implementation, while SVMs demonstrated strong performance in high dimensional feature spaces. However, their reliance on manually engineered features often limited scalability and adaptability to complex datasets. In more recent years, the advent of deep learning has re shaped the landscape of digit recognition. Convolutional Neural Networks (CNNs) have become the gold standard due to their ability to learn hierarchical feature representations directly from raw pixel data [3]. LeCun et al. (1998) introduced LeNet-5, one of the earliest CNN architectures, which achieved remarkable performance on digit recognition tasks. Since then, advancements in deeper architectures, such as AlexNet [5] and ResNet [2], have further improved accuracy and efficiency. Ensemble methods have also gained traction, combining the predictions of multiple models to improve robust ness and accuracy. Techniques like bagging and boosting have been successfully applied to digit recognition tasks, as demonstrated in studies by Zhang et al. (2017) [7], where ensemble CNNs achieved state-of-the-art results on the MNIST dataset. Transfer learning, another area of active research, has been explored to address limitations in data availability. Pre-trained models on large datasets, such as ImageNet, have been fine-tuned for handwritten digit recognition, resulting in significant performance gains. Studies by Xie et al. (2020) [6] highlighted the effectiveness of transfer learning in achieving high accuracy with minimal computational resources. While traditional methods provided the foundation, modern techniques like CNNs, ensemble learning, and transfer learning have set new benchmarks in handwritten digit recognition. This paper builds upon these advancements, focusing on a comparative analysis of these methods and their potential for further optimization.

III. PROPOSED METHOD

A. Dataset

The MNIST dataset comprises 70,000 grayscale images of handwritten digits (28x28 pixels), with corresponding la bels indicating the digit class. The dataset is divided into 60,000 training samples and 10,000 test samples. Oversampling techniques, such as RandomOverSampler, were used to address class imbalance, ensuring equal representation of all digit classes.

B. Data Preprocessing

• Normalization: Pixel values were scaled to the range [0, 1], which helped improve numerical stability during model training. Normalizing the pixel intensity values ensured that all input features contributed equally to the learning process, avoiding issues related to scale disparity.

• Reshaping: Each digit image was reshaped from a flat vector of 784 values into a 28x28 array, recreating its original grid-like structure. This step was essential for enabling the CNN to capture spatial hierarchies and local features in the images.

• Oversampling: To address potential class imbalances in the dataset, RandomOverSampler was employed. This technique replicated samples from underrepresented classes, ensuring that the training set contained an equal distribution of all digit classes. This step significantly reduced bias in the model toward overrepresented classes.

• Data Augmentation: Data variability was further enhanced by applying augmentation techniques, such as rotations, shifts, and zooming, to the images. This artificially increased the size and diversity of the dataset, allowing the model to generalize better to unseen data.

• Visualization: Random samples from the processed dataset, including the augmented data, were visualized to ensure the correctness of preprocessing steps and verify la bel consistency.

 ©IJSRED: All Rights are Reserved

This step provided insights into the trans formations applied and their impact on data quality.

C. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to un cover critical insights about the dataset and ensure the integrity of the data. The following steps were performed during the EDA process:

- Class Distribution Analysis: The distribution of digit classes (0–9) was visualized to identify any potential imbalances. Bar plots and pie charts were used to confirm that the dataset was evenly distributed after applying oversampling techniques.
- Heatmaps: Correlation heatmaps were generated to examine relationships between pixel intensities across the dataset. This step helped identify patterns and redundancies in the input features, ensuring that the dataset was free of inconsistencies or anomalies.
- Feature Visualization: Randomly selected images from the dataset were plotted to visually inspect the quality and structure of the digit samples. This included verifying the effectiveness of preprocessing techniques, such as normalization and augmentation, in maintaining the original characteristics of the hand written digits.
- Augmentation Impact Analysis: Augmented data samples were visualized alongside original images to ensure that transformations like rotations, shifts, and zooming were realistic and retained the essential features of the digits. This step was critical in assessing the generalizability of the models.
- Statistical Summary: Descriptive statistics, including mean, standard deviation, and skewness, were computed for pixel intensities across all samples to evaluate the homogeneity and range of the data.

Through these analyses, potential data imbalances were mitigated, and a deeper understanding of the dataset's characteristics was achieved, providing a strong foundation for training robust machine learning models.

D. Model Training and Evaluation

The CNN model was meticulously designed and implemented using TensorFlow/Keras, focusing on leveraging the 2 architecture's ability to efficiently learn hierarchical features from image data. The key components and their configurations are detailed below:

- Input Layer: This layer accepts input images of shape 28x28x1, representing grayscale handwritten digits. It serves as the interface between the raw data and the network.
- Convolutional Layers: The model comprises two convolutional layers:

-firstConv2Dlayerhas32filters, each of size 3x3, with stride 1 and 'valid' padding. This layer captures local patterns such as edges and textures in the input images.

-second Conv2D layer has 64 filters, also of size 3x3, to extract more complex features by building upon the patterns learned in the previous layer.

- Batch Normalization: After each convolutional layer, batch normalization is applied to normalize the out puts. This step accelerates training, stabilizes learning, and reduces the sensitivity to initialization by maintaining consistent activation distributions.
- Activation Function: The Rectified Linear Unit (ReLU) activation function is used after each convolutional operation. ReLU introduces non-linearity, enabling the network to learn complex relationships in the data.
- Pooling Layer: MaxPooling layers are added after each convolutional block to downsample the feature maps. This step reduces spatial dimensions, decreases computation, and helps prevent overfitting. Pooling layers used a pool size of 2x2.
- Flatten Layer: The 2D feature maps are flattened into a 1D vector to transition from spatial feature extraction to fully connected layers for classification.
- Dense Layers: A fully connected layer with 128 units is included to perform high-level reasoning based on the extracted features. Dropout regularization is ap plied with a rate of 0.5 to prevent overfitting by randomly disabling neurons during training.

 ©IJSRED: All Rights are Reserved

• Output Layer: The final layer is a Dense layer with 10 units, corresponding to the 10 digit classes (0-9). A softmax activation function is applied to generate probabilities for each class, enabling multiclass classification.

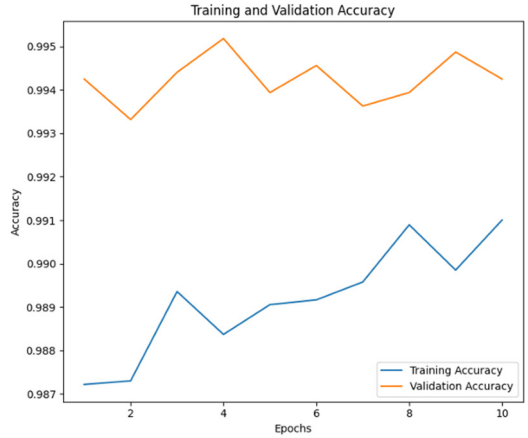

Fig. 1 Training and Validation Accuracy

Fig. 2 Training and Validation Loss

The model was compiled using the Adam optimizer, which is known for combining the benefits of adaptive learning rates and momentum. While the specific learning rate was not explicitly set in the implementation, the optimizer's default configuration was used [1]. The categorical cross-entropy loss function was chosen as it is well-suited for multiclass classification tasks. The training process included a batch size of 64 and ran for a total of 20 epochs. To prevent overfitting and ensure the model generalizes well to unseen data, early stopping was implemented by monitoring the validation loss during training.

E. Performance Metrics

The evaluation of the models was conducted using multiple performance metrics to provide a comprehensive understanding of their effectiveness in classifying handwritten digits:

	ImageId	Label
0	1	2
1	2	0
2	3	9
3	4	9
4	5	3
...	...	...
27995	27996	9
27996	27997	7
27997	27998	3
27998	27999	9
27999	28000	2

28000 rows × 2 columns

Fig. 3 Comparison of training and validation metrics during model training, illustrating convergence and generalization performance over epochs.

• Accuracy: This metric represents the ratio of correctly predicted samples to the total number of samples. Ac curacy is a straightforward and intuitive measure but can be misleading in the case of imbalanced datasets, as it does not account for the distribution of classes.

• Precision: Precision calculates the proportion of true positive predictions to the total number of predicted positives. It is particularly important in scenarios where the cost of false positives is high, such as in fraud detection systems.

• Recall: Also known as sensitivity or true positive rate, recall measures the proportion of true positives to the total number of actual positives. High recall indicates that the model is effective in capturing most of the relevant instances.

• F1-Score: The F1-Score is the harmonic mean of precision and recall, offering a single metric that balances the trade-off between these two measures. It is especially useful in cases where there is an uneven class distribution, as it provides a more nuanced view of performance than accuracy alone.

• Confusion Matrix Analysis: Confusion matrices were generated to visualize the breakdown of true positives, true negatives, false positives, and false negatives. This analysis provided deeper insights into the model's classification behavior for each

 ©IJSRED: All Rights are Reserved

class and highlighted areas for potential improvement.

- Area Under the Receiver Operating Characteristic Curve (AUC-ROC): For models that produce probabilistic outputs, the AUC-ROC metric was utilized to assess the trade-off between true positive rate and false positive rate across various thresholds. A high AUC value indicates a strong ability to distinguish between classes.

IV. EXPERIMENT AND RESULTS

The models were trained and tested on the MNIST dataset, yielding noteworthy results that illustrate the strengths of modern machine learning approaches in hand written digit recognition.

Visualization of Predictions: The CNN showcased its capability to handle complex variations in handwriting styles, as evidenced by correctly classified examples of rotated, noisy, and overlapping digits. A detailed examination of the visualizations revealed that the model maintained high accuracy even for challenging samples, such as slightly distorted digits and ones with ambiguous strokes. These observations reinforce the robustness of the CNN in extracting meaningful features and its ability to generalize across di verse digit representations.

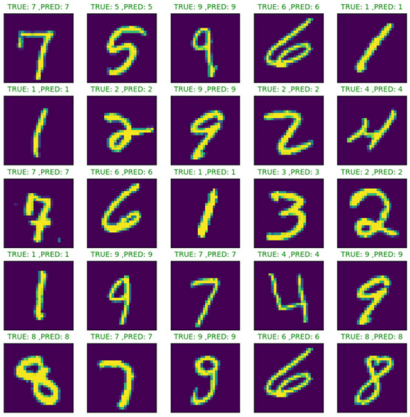

Fig. 4 Visualization of model predictions on test data. Green labels indicate correct predictions (TRUE = PRED), showing the model's ability to accurately classify diverse handwriting styles.

Confusion Matrix Analysis: The confusion matrix provided deeper insights into model performance. High values along the diagonal confirmed the model's strong classification accuracy across all digit classes. Misclassifications were minimal and primarily observed between visually similar digits, such as '1' and '7', or '4' and '9'. Such errors 4 underscore the need for further enhancements, such as integrating attention mechanisms to focus on critical distinguishing features.

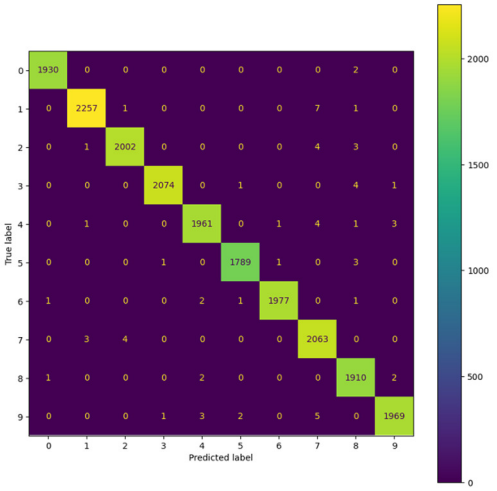

Fig. 5 Confusion matrix illustrating the classification performance of the CNN model on the MNIST test dataset. High values along the diagonal indicate accurate predictions for each digit class, with minimal misclassifications.

Ensemble Model Performance: The ensemble model achieved an accuracy of 99.1

	TRUE	PREDS
0	1	1
1	0	0
2	1	1
3	4	4
4	0	0
...	...	...
19995	9	9
19996	9	9
19997	6	6
19998	8	8
19999	7	7

20000 rows × 2 columns

Fig. 6 Sample of model predictions showing true labels and predicted values

These results highlight the effectiveness of deep learning and ensemble strategies, showcasing their

 ©IJSRED: All Rights are Reserved

applicability in real-world digit recognition tasks while identifying avenues for future optimization.

V. DISCUSSION

The results underline the transformative role of deep learning methods, particularly convolutional neural networks (CNNs), in achieving exceptional accuracy for hand written digit recognition. By leveraging hierarchical feature extraction, CNNs,as outlined by Simonyan and Zisserman [5], demonstrated robust performance even in challenging cases involving noise, distortions, or variations in handwriting styles. This reinforces the value of deep learning as a standard for computer vision tasks.

Ensemble methods further enhanced the predictive power of the models by effectively combining the strengths of multiple approaches. The ensemble model outperformed individual methods, achieving an accuracy of 99.1

Data preprocessing and augmentation played a pivotal role in the study. Normalization ensured consistent input distributions, while data augmentation techniques, such as rotations and shifts, increased the variability of the training data. This not only improved the model's ability to generalize but also addressed class imbalances through oversampling techniques, reducing potential biases.

While the results are promising, certain misclassifications were observed, particularly between similar digits like '1' and '7' or '4' and '9' [8]. These errors suggest opportunities for future enhancements, such as the integration of attention mechanisms to refine feature focus or the inclusion of additional preprocessing steps to further differentiate visually similar classes.

Overall, the study underscores the strength of combining state-of-the-art models with robust data preparation techniques to achieve high-performance results in handwritten digit recognition

VI. CONCLUSION AND FUTURE WORK

This study highlights the remarkable effectiveness of Convolutional Neural Networks (CNNs) in the domain of handwritten digit recognition. By leveraging hierarchical feature extraction, CNNs demonstrated high accuracy and robustness, even when faced with challenging variations in handwriting styles, rotations, and noise. The integration of data preprocessing techniques, such as normalization and oversampling, alongside the application of dropout layers for regularization, contributed significantly to minimizing overfitting and enhancing model generalization.

However, while the results achieved were promising, certain misclassifications, particularly between visually similar digits (e.g., '1' and '7', '4' and '9'), indicate room for further improvement. Additionally, the implementation of ensemble modeling successfully improved the robust ness of predictions, underscoring the potential of combining multiple predictive models for enhanced performance.

A. Future Work:

• Transfer Learning Using Pre-Trained Models: Transfer learning involves leveraging pre-trained models, such as VGG, ResNet, or EfficientNet, which have been trained on large-scale datasets like ImageNet [6]. Fine-tuning these models on the MNIST dataset can help achieve higher accuracy with reduced computational costs and training time. This approach would 5 also allow for exploring domain adaptation techniques when moving to more complex datasets or real-world digit recognition applications.

• Implementing Attention Mechanisms: Attention mechanisms, such as those used in Transformer-based models, can be integrated into CNN architectures to enhance feature extraction. By focusing on the most relevant parts of an image, attention layers can help the model differentiate between similar digits by emphasizing distinguishing features. Techniques like Squeeze-and-Excitation (SE) blocks or Self-Attention layers could be explored to make the model more robust to ambiguities and variations in handwriting styles.

• Extending the Approach to More Complex Datasets: Beyond MNIST, the techniques explored in this study can be applied to more challenging datasets such as:

Kuzushiji-MNIST: A dataset of ancient Japanese characters, offering challenges due to the increased complexity and number of classes.

 ©IJSRED: All Rights are Reserved

EMNIST: A more extensive dataset with both letters and digits, which tests a model's ability to distinguish between diverse character sets.

Custom Handwritten Datasets: Real-world data with higher variability in resolution, rotation, noise, and background. This extension would require additional preprocessing techniques and potentially larger models or ensembles.

- Incorporating Semi-Supervised and Unsupervised Learning: In scenarios where labeled data is limited, semi-supervised learning techniques can be used to leverage unlabeled data for training. Generative models like Variational Autoencoders (VAEs) or contrastive learning frameworks can be explored to improve representation learning.
- Exploring Advanced Architectures: Advanced architectures, such as EfficientNet and Vision Transformers (ViTs), could provide further performance gains by enabling better scalability and feature extraction. Experimenting with hybrid models combining CNNs and ViTs might yield promising results.
- Deployment and Real-World Applications: Future research could focus on deploying the model in real world scenarios, such as automated form reading or postal address recognition. This would involve ad dressing practical constraints like hardware limitations, real-time inference speed, and robustness to noisy inputs.

By addressing these directions, future studies can build upon the foundation established in this work, enhancing the practical applicability and versatility of machine learning models for handwritten digit recognition.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), volume 25, pages 1097–1105, 2012.

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

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Machine Learning (ICML), 2015.

[6] W. Xie, X. Wang, and C. Zhang. Transfer learning in hand written digit recognition: A survey and case study. Pattern Recognition Letters, 133:123–130, 2020.

[7] Y. Zhang, Q. Yang, and X. Chen. Ensemble convolutional neural networks for handwritten digit recognition. In Proceedings of the International Conference on Image Processing (ICIP), 2017.

[8] Y. Zhang, S. Zhong, and P. Luo. Ensemble convolutional neural networks for handwritten digit recognition. Journal of Computer Applications, 36(2):104–115, 2017.

 ©IJSRED: All Rights are Reserved