# Handwritten English Digit Recognition

Mr. Akshay Nayak B.S [1], Abhijna R S [2], Dashami H N [3], Hemanth K S [4], Hrudai Pavankumar S P [5]
*Department of Computer Science and Engineering*
*Jawaharlal Nehru New College of Engineering, Shivamogga, Karnataka, India.*
akshaynyk@jnnce.ac.in [1], rsabhijna2004@gmail.com[2], dashamihn123@gmail.com [3],kshemanth788@gmail.com[4], sphrudaypavankumar@gmail.com[5]

***Abstract*—Handwritten digit recognition has become a key area of research in the fields of computer vision,machine learning and pattern recognition. This project focuses on development of robust model for handwritten English digit recognition implementing CNN, a class of the deep learning algorithms particularly well-suited for image pro- cessing tasks. The model is trained and judged on the MNIST dataset, which consists of grayscale images of handwritten digits from 0 to 9. The CNN architecture employed in this project incorporates convolutional layers for the feature extraction, pooling layers for the dimension reduction and connected layers for the classification. Through extensive experimentation and hyperparameter tuning, the model achieves a high accuracy in digit recognition, demonstrating the effectiveness of CNNs in learning spatial hierarchies of features from input images. This system can be applied in areas such as postal automation, bank check processing, and digitization of handwritten forms.**

***Index Terms*—Handwritten Digit Recognition, Random Forest, SVM, Graph Neural Networks, Chain Codes, MNIST,Feature extraction, Machine Learning, CNN**

## I. Introduction

The Handwritten English Digit Recognition (HDR) is a foundational problem of artificial intelligence and computer vision, focusing on identifying digits (0–9) written in various human handwriting styles. Despite appearing simple, HDR is challenging due to variations in stroke. Its significance spans numerous applications, including automated cheque processing in banking, zip code recognition in postal services, evaluation of handwritten exam papers in education, and digitization of patient records in healthcare. Early approaches to HDR relied on rule-based and statistical techniques such as pixel matching, template comparison, geometric feature extraction, and projection histograms. While these methods laid the groundwork, they were limited by their dependence on handcrafted features and poor adaptability to diverse handwriting. The introduction of machine learning algorithms like K-Nearest Neighbours (KNN), Support Vector Machines (SVM), and Decision Trees offered better generalization, but still treated images as flattened vectors, ignoring spatial relationships. A major advancement came with the advent of deep learning algorithm, particularly Convolutional Neural Networks (CNNs). CNNs mimic the hierarchical processing of human system, enabling automatic learning of spatial features from raw pixel data. They use convolutional filters, pooling layers, and activation functions to detect patterns at multiple levels of abstraction. Their architectural strengths—local connectivity, weight sharing, and efficient parameter usage—make CNNs highly robust to distortions and computationally efficient. The MNIST dataset, comprising training images of 60,000 and testing images of 10,000 of handwritten digits, has been pivotal in the development of HDR systems. It served as a standard benchmark that accelerated the adoption of deep learning, with CNN-based models consistently outperforming traditional techniques. Today, HDR not only demonstrates the evolution of pattern recognition but also serves as a gateway to more complex visual recognition tasks.

## II. Literature Survey

### A. *Random Forest-based Digit Recognition*

Bernard et al. [1] explores the use of Random Forest (RF) classifiers on the MNIST dataset. The Random Forest approach leverages the ensemble learning technique, merging multiple decision trees trained on bootstrap samples with randomized feature selection at each node. Their method, Forest-RI, demonstrates high accuracy and robustness to overfitting. The authors specifically tune parameters like the number of trees (L) and feature subset size (K), revealing optimal performance with modest tree counts. The study confirms Random Forest as a competitive classifier when compared to Boosting and Bagging.

### B. *Graph Neural Network (GNN) Approach*

Sharma et al. [2] propose a novel technique integrating Graph Neural Networks (GNN) with chain code-based feature extraction for recognizing handwritten trajectories. Unlike traditional models, this approach transforms the handwriting trajectory into a graph structure, where nodes represent pixel segments and edges define their sequential connections. Both offline (using drawing order recovery) and online handwriting datasets are employed. Their results on MNIST and Unipen datasets show significant accuracy improvements, with GNN outperforming conventional SVM and HMM models in fewer training epochs. This hybridization of geometric and deep learning methods is computationally intensive but promising in performance.

### C. SVM-based Digit Classification

Reddy et al. [3] presents a digit recognition system using Support Vector Machines (SVM). The system employs standard preprocessing steps-grayscale conversion, binarization, and contour-based feature extraction followed by SVM classification. Polynomial kernels are found to be effective for the high-dimensional MNIST dataset. Achieving a test accuracy of 97.83%, the model validates SVM's strength in linear and non-linear classification. Although deep learning technique have largely superseded SVMs, this technique remains familiar for its simplicity and interpretability, particularly in resource-constrained settings.

### D. PCA-Based Logistic Regression

Chao-Chung Peng. [4] introduced a robust approach to digit recognition using PCA combined with Logistic Regression, particularly optimized for rotated digits in consumer electronics. The system per- forms image straightening using PCA, noise reduction via convolution filters, and dimensionality reduction before clas- sification. The method significantly reduces computational load and memory usage, outperforming traditional logistic regression with an 18.5% improvement. It showed high real- time performance even on low-power devices and reached 77.5% accuracy on rotated MNIST images, making it suitable for embedded applications.

### E. Edit Distance-Based KNN

Bernard et al. [5] proposed a novel digit recognition technique based on edit distance and K-Nearest Neighbors (KNN). Digits are encoded as Freeman chain codes, transforming image data into sequential strings. The classifier measures similarity using edit distance and is enhanced through data cleaning techniques like removing outliers and irrelevant examples. This approach, while computationally heavier, benefits from interpretability and simplicity. The system, targeted at student projects, demonstrates respectable accuracy and real-time performance through dataset reduction and nearest-neighbor search optimizations.

### F. Convolutional Neural Networks and Their Variants

Feiyang Chen et al. [6] provided a comparative study of several CNN-based models including standard CNN, ResNet, DenseNet, and Capsule Networks (CapsNet) on the MNIST dataset. Traditional CNNs, while powerful, lose spatial hierarchies due to pooling layers. ResNet addressed degradation issues in deeper networks through residual connections. DenseNet further improved information flow via dense connectivity. However, CapsNet demonstrated the highest accuracy (99.75%) with fewer data and better generalization. CapsNet's dynamic routing and vector-based capsules preserved positional and orientation information, outperforming other models in both accuracy and data efficiency.

### G. Naive Bayesian Approach

Wang and Zhang et al. [7] presented a modified Naive Bayes classifier tailored to high-dimensional data. Traditional naive Bayesian inference suffers from numerical underflow in high dimensions due to product of small probabilities. The authors propose using geometric means instead of direct multiplication to stabilize the probabilities. This allows for meaningful secondary predictions beyond the most probable class, thus enhancing classification quality. Evaluated on the MNIST dataset, this technique not only avoided underflow but also preserved accuracy and interpretability in multiclasses.

### H. Pytesseract

Dr. K Soumya et al. [8] traces the evolution of OCR and highlights its growing utility in digitizing handwritten documents. Lefe`vre and Piantanida formalized the introduction of Pytesseract, making OCR capabilities accessible through Python for developers and researchers.

The methodology behind Pytesseract includes:

- Preprocessing: Grayscale conversion, Gaussian blur, and thresholding using OpenCV to enhance OCR readiness.
- OCR Execution: Pytesseract applies recognition models to extract text, with options to tune engine and page segmentation modes.
- Post-processing: Enhancements such as grammar correction, spelling check, case normalization, and duplicate word removal improve readability and reduce error.

### I. Logistic Regression with SGD

Chen et al. [9] investigate a binary classification approach to digit recognition using stochastic logistic regression. The study focuses on distinguishing digit pairs (0 vs. 1 and 3 vs. 5) from the MNIST dataset. The parameters are estimated via stochastic gradient descent (SGD), which updates model weights per sample to accelerate convergence. The results show test accuracies of 97–99% for 0 vs. 1 and 85–94% for 3 vs. 5 in various sample sizes.

The method is notable for its robustness even with the short training data. The Analysis confirms that performance plateaus after 30% of the data, suggesting a strong generalization due to the assortment of MNIST. Although simple, interpretation of the trained model demonstrates the efficacy of SGD for logistic regression in real-world classification scenarios, especially for binary problems.

### J. Multiclass Logistic Regression with Feature Engineering

Ahmad et al. [10] present a comprehensive study using multiclass logistic regression for digit classification in all ten MNIST classes. The system implements extensive preprocessing, such as normalization, one-hot encoding, and pixel vector flattening, followed by training a logistic regression model using gradient descent. Performance evaluation includes metrics such as accuracy, confusion matrices, and F1 scores, with the model achieving around 90% accuracy.

TABLE I
SUMMARY OF TECHNIQUES USED IN HANDWRITTEN DIGIT RECOGNITION

| Reference / Author | Method / Approach | Key Features / Techniques | Findings / Inference |
|---|---|---|---|
| Bernard et al. (2007) | Random Forest Classifier | Bootstrap sampling, randomized feature selection, parameter tuning (L, K) | Achieved high accuracy and robustness; competitive with Boosting/Bagging |
| Sharma et al. (2024) | Graph Neural Network (GNN) | Chain code transformation, graph-based pixel encoding, spatial trajectories | Outperformed SVM/HMM; effective on MNIST and Unipen with fewer epochs |
| Reddy et al. (2022) | Support Vector Machine (SVM) | Polynomial kernel, contour-based features, grayscale preprocessing | Achieved 97.83% accuracy; useful for resource-limited environments |
| Peng et al. (2023) | PCA + Logistic Regression | PCA for rotation correction, dimensionality reduction, convolution filters | 77.5% accuracy on rotated digits; real-time on low-power devices |
| Bernard et al. (2012) | Edit Distance-Based KNN | Freeman chain code encoding, edit distance metric, data cleaning | Simple yet interpretable; effective real-time performance with optimizations |
| Chen et al. (2019) | CNN, ResNet, DenseNet, CapsNet | Comparative study, pooling, residual and capsule layers | CapsNet achieved 99.75% accuracy; best generalization on low data |
| Wang and Zhang (2019) | Naive Bayesian with Geometric Mean | Probability stabilization, handling numerical underflow | Preserved accuracy in high-dimensions; improved multi-class interpretability |
| Soumya et al. (2024) | OCR with Pytesseract | OpenCV preprocessing, grammar/spell correction, post-OCR enhancement | Effective digit recognition from handwriting; high real-world utility |
| Chen et al. (2018) | Logistic Regression with SGD | Binary classification (0 vs 1, 3 vs 5), convergence with stochastic gradient | 85–99% accuracy; strong generalization even with 30% training data |
| Ahmad et al. (2025) | Multiclass Logistic Regression | One-hot encoding, feature extraction (gradient, concavity, radial) | 90% accuracy; good for standard hardware, interpretable ML approach |

## III. RECOMMENDATIONS AND FUTURE DIRECTION

This study demonstrates that CNN-based models are highly effective for handwritten digit recognition, yet there is considerable scope for enhancement. Future work can explore advanced architectures like Capsule Networks and Vision Transformers to better capture spatial hierarchies and contextual dependencies. Combining deep learning with graph-based or probabilistic models may also improve accuracy, especially on noisy or overlapping inputs. Expanding the dataset to include multilingual, cursive, or real-world samples would test model robustness in practical scenarios. Furthermore, deploying optimized lightweight models on edge devices can enable offline recognition for mobile or embedded systems. Real-time feedback, adaptive learning for user-specific patterns, and interpretable AI are also promising areas to ensure scalability, fairness, and ethical usage in high-stakes domains.

## IV. CONCLUSION

Each method surveyed contributes uniquely to the task of handwriting digit recognition. Random forests provide a reliable baseline with minimal tuning. SVMs, with their mathematical rigor, offer competitive performance and ease of deployment. GNNs emerge as a frontier technique for structured data, particularly when enriched with trajectory or spatial information. In future research hybrid models can be explored and graph-based enhancements to combine the strengths of each approach. Logistic regression, though foundational, remains relevant for handwritten digit recognition. Its strengths lie in simplicity, transparency, and reliable performance for moderately complex tasks. Chen's work illustrates its suitability for binary problems with limited data, while Ahmad et al. show how pre-processing and feature engineering can extend it to multiclass classification. Future research can explore combining logistic regression with neural layers or ensemble techniques to further boost accuracy while maintaining interpretability. The application of Pytesseract to handwritten text recognition illustrates the potential of open-source tools in achieving high OCR accuracy. The preprocessing and post-processing strategies, makes it suitable for real-world tasks like archiving, data entry au- tomation, and accessibility technologies. Pytesseract continues to evolve, offering a flexible platform for developers working in document recognition and analysis.

## REFERENCES

[1] S. Bernard, L. Heutte, and S. Adam, " Random Forests for Handwritten Digit Recognition," *Proc. of ICDAR*, 2007.
[2] A. Sharma, S. Singh, and S. Ratna, "Graph Neural Network Based Handwritten Trajectories Recognition," *arXiv preprint*, 2024.
[3] B. P. K. Reddy et al., "Handwritten Digit Recognition Using SVM Algorithm in Machine Learning," *IJJCRT*, vol. 10, no. 6, 2022.
[4] Chao-Chung Peng, Chao-Yang Huang, Yi-Ho Chen, "Principal Component Analysis-Based Logistic Regression for Handwritten Digit Recognition in Consumer Devices," *Electronics*, 2023.
[5] Marc Bernard et al. "Handwritten Digit Recognition using Edit Distance-Based KNN," Teaching Machine Learning Workshop, Edinburgh, 2012.
[6] Feiyang Chen, Nan Chen, Hanyang Mao, Hanlin Hu "Assessing Four Neural Networks on Handwritten English Digit Recognition Dataset (MNIST)" 20 Jul 2019.
[7] Kai Wang, "A Novel Naive Bayesian Approach to Inference with Applications to the MNIST Handwritten Digit Classification."
[8] Dr. K Soumya et al., "Handwritten Text Recognition Using Pytesseract" Vol. 14 Issue 04, April 2024.
[9] A. W. Chen, "Handwriting Recognition and Prediction Using Stochastic Logistic Regression," *Int. J. Inf. Res. Rev.*, vol. 5, pp. 5526–5527, 2018.
[10] F. Ahmad et al., "Handwritten Digit Recognition Using Logistic Regression," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 12, pp. 716–722, 2025.