Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20250402051
Volume 4, Issue2 (May-August 2025), PP: 371-374.
www.indjcst.com
ISSN No: 2583-5300

Identifying Handwritten Recognition Using Logistic Regression in Pytorch

Chaithanya H[1], Sai Sumanth C[2], Thrisha N[3]
[1, 2, 3] Department of CSE - Data Science, Dayananda Sagar Academy of Technology and Management, Bengaluru, Karnataka, India.

To Cite this Article: *Chaithanya H[1], Sai Sumanth C[2], Thrisha N[3], "Identifying Handwritten Recognition Using Logistic Regression in Pytorch", Indian Journal of Computer Science and Technology, Volume 04, Issue 02 (May-August 2025), PP: 371-374.*

Abstract: *Handwritten digit recognition is essential in applications like postal sorting, cheque verification, and document digitization. While convolutional neural networks (CNNs) achieve high accuracy in image classification, their complexity and resource demands limit their use in low-power or transparency-focused environments. This study compares a classical logistic regression model with a CNN for digit classification on the MNIST dataset using PyTorch. The MNIST dataset consists of 60,000 training and 10,000 test grayscale images (28×28 pixels). Logistic regression flattens each image into a 784-dimensional vector and applies a linear classifier optimized with cross entropy loss and stochastic gradient descent. The CNN, on the other hand, uses convolutional layers to automatically extract spatial features for classification. Experimental results show logistic regression achieves over 90% accuracy, demonstrating efficiency and interpretability. Meanwhile, the CNN surpasses 98% accuracy, highlighting the advantage of deep learning for complex feature extraction. This comparison underscores the trade-off between model simplicity and performance. Logistic regression serves as a strong baseline or practical solution when resources and interpretability are priorities, while CNNs offer superior accuracy for more demanding tasks.*

Key Words: *Handwritten Digit Recognition, Logistic Regression, Convolutional Neural Network, MNIST Dataset, Image Classification, PyTorch, Cross-Entropy Loss, Stochastic Gradient Descent, Model Interpretability, Computational Efficiency, Deep Learning, Baseline Model, Feature Extraction, Accuracy.*

I.INTRODUCTION

The rapid growth advancement of digital technologies across various industries, there has been a notable increase in the need for precise and automated systems that can identify handwritten digits. Although this task may seem straightforward, it is a foundational challenge in machine learning and has historically been a standard for assessing the effectiveness of classification algorithms. In recent years, deep learning methods, especially convolutional neural networks (CNNs), have taken the lead in this area due to their remarkable accuracy and capability to automatically discern intricate patterns.

Nonetheless, simpler algorithms like logistic regression still possess considerable merits. Their advantages include computational interpretability, demands, and minimal ease of implementation, making them particularly suitable for rapid prototyping, situations with limited resources, or educational applications.

This project investigates the creation of a handwritten digit recognition system utilizing logistic regression, developed in PyTorch — a flexible, open-source deep learning framework celebrated for its approachable syntax and dynamic computation graph. The emphasis is on employing logistic regression as a baseline classification model, applied to the MNIST dataset, which comprises 70,000 grayscale images of handwritten digits from 0 to 9.

The project encompasses the entire workflow: from data pre-processing and feature engineering to model design, training, and assessment. By systematically navigating these steps, the project illustrates how even conventional machine learning techniques can remain pertinent and effective in today's AI environment. Moreover, leveraging PyTorch not only simplifies the development process but also offers a robust foundation for future exploration of more sophisticated models.

II.LITERATURE SURVEY

A. "Evolutionary CNNs for Handwriting Recognition – (Neurocomputing, Baldominos et al". 2018)

This research introduces an automated approach to designing convolutional neural networks (CNNs) using neuroevolution techniques. By leveraging genetic algorithms and grammatical evolution, the architecture and hyper parameters of CNNs are optimized without manual intervention. The method enables learning directly from raw image inputs, bypassing the need for handcrafted features or extensive preprocessing. When applied to the MNIST dataset, the evolved CNNs achieved competitive performance, highlighting the effectiveness of evolutionary strategies in deep learning model development.

B. "Neural Network-Based Digit Recognition Prototype – Anonymous" (Year Not Specified)

This study outlines a prototype system for recognizing handwritten digits through neural networks and image processing. The process involves digitizing handwriting, segmenting characters into 16x16 pixel grids, and training a three-layer neural network to recognize the patterns. This method diverges from conventional techniques by mimicking cognitive behaviour rather than relying solely on pixel-level analysis. The result is a lightweight and efficient model capable of real-time application in areas like biometrics and education.

C. "Handwriting Feature Dataset for Statistical Analysis – Agius et al". (Data in Brief, 2018)

The authors provide a specialized dataset that encodes various structural and spatial handwriting features for use in statistical modelling. The dataset includes samples from native English writers in Australia and Vietnam, with the goal of supporting forensic analysis and country-of-origin profiling. This dataset aids comparative studies and offers a foundation for advanced research in writer identification and document forensics.

D. "Machine Learning Models for Signature Verification – Akter et al". (APCIT, 2024)

This paper compares the performance of several machine learning approaches, including CNNs, SVMs, and hybrid systems, in verifying handwritten signatures. Using publicly available datasets, the study finds CNNs to be the most reliable in both accuracy and resistance to forgery attempts. The work underlines the importance of robust models in biometric verification systems and provides insights for developing secure identity authentication technologies.

E. "Detecting Confusing Drug Names Using Evolutionary Logistic Regression – Millán Hernández et al". (2019)

A novel logistic regression model enhanced through evolutionary algorithms is proposed to identify drug names that appear or sound similar, a known cause of prescription errors. Unlike traditional methods limited to a few similarity metrics, this approach uses 21 different measures, optimized via genetic algorithms, to improve classification on imbalanced datasets. The model significantly boosts detection accuracy, making it highly applicable in pharmaceutical safety and healthcare information systems.

F. "Survey on On-Line Handwriting Recognition – Tappert et al". (IEEE TPAMI, 1990)

This seminal review explores the development of real-time handwriting recognition technologies, discussing key elements such as input digitizers, shape-matching algorithms, and signal processing techniques. The paper differentiates on-line systems—which capture dynamic input like stroke order and pressure—from off-line recognition. Though published decades ago, it remains a foundational reference for researchers exploring the early innovations in pen-based computing and handwriting interfaces.

G. "Hybrid Machine Learning Model for Customer Segmentation – Arakh et al". (IJSDR, 2024)

This study proposes a customer segmentation framework that combines unsupervised clustering with supervised classification to improve segmentation outcomes. The hybrid model incorporates techniques like K-Means, Support Vector Machines (SVM), and Decision Trees, enabling more precise groupings by refining clusters with labelled data and domain knowledge. The system proved effective in adapting to evolving consumer behaviors and supports personalized marketing strategies to enhance customer retention and business performance. Correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

III.METHODS

The methodical strategy employed to design, train, and assess a machine learning model for recognizing handwritten digits using logistic regression. The objective was to investigate the effectiveness of a straightforward yet interpretable model utilizing the MNIST dataset and to scrutinize its performance in classifying images of grayscale digits. The methodology consists of five primary components: selection of the dataset, data preprocessing, architecture of the model, process of training, and evaluation of performance.

Selection of Dataset: The MNIST Benchmark For this research, the MNIST (Modified National Institute of Standards and Technology) dataset was chosen as the basis. It is a well-established benchmark in image classification, particularly for evaluating models on tasks related to recognizing handwritten digits. The dataset includes 70,000 grayscale images, divided into 60,000 for training and 10,000 for testing purposes. Each image is a square of 28×28 pixels depicting digits from 0 to 9. MNIST is particularly beneficial due to its cleanliness, structured format, and standardization. Each image features a single digit, centred and normalized in size, making it ideal for preliminary experimentation and comparison of machine learning models without the added complexities found in real world noise.

Data Pre-processing: Cleaning and Preparing Inputs Prior to inputting the data into the model, several important pre-processing steps were required to transform the raw pixel data into a format amenable to logistic regression.

- Flattening the Image: Because logistic regression is a linear model that functions with vectorized inputs, the 28×28 images were flattened into one-dimensional arrays comprising 784 features. This transformation allows each image to be treated as a single input vector, where each element represents a pixel's intensity. Normalization: The original pixel values vary from 0 to 255. To standardize all features and enhance training efficiency, pixel intensities were normalized to a range of [0, 1] by dividing each value by 255. Normalization aids in preventing large gradients and facilitates faster convergence during training. Label Encoding: The labels in the MNIST dataset consist of digits ranging from 0 to 9.

For multi-class classification, each label remains as an integer class. PyTorch's Cross Entropy Loss directly handles these class labels, meaning that one-hot encoding is unnecessary. These pre-processing steps are vital for streamlining the learning process, enhancing numerical stability, and ensuring alignment with the logistic regression model.

Model Architecture: A Simple Linear Classifier The architecture of the model employed in this research is a logistic regression classifier, realized as a single-layer neural network. Despite its simplicity, logistic regression can act as an effective baseline for classification tasks when the data is linearly separable.

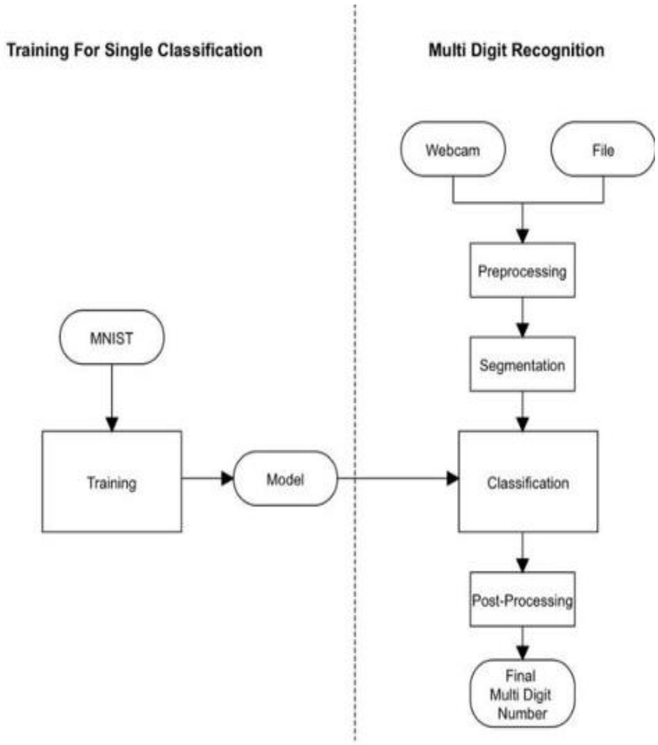

Figure 1: Identifying hand written recognition using logistic regression in Py Torch flow chat

• **Input Layer**: Takes in a vector of 784 dimensions (the flattened image).

• **Output Layer:** Produces 10 values, each corresponding to a digit class (0–9). These are normalized scores, or logits.

• **Softmax Activation:** While not explicitly applied in the model's definition, Py Torch's Cross Entropy Loss internally integrates the softmax function with negative log-likelihood loss. Soft max transforms log its into probability distributions across the classes.

• **Loss Function:** Cross Entropy Loss served as the objective function. It assesses the predicted class probabilities against the actual labels and following the training phase, the model's penalizes the model based on the confidence of incorrect predictions.

• **Optimizer:** The model was trained utilizing Stochastic Gradient Descent (SGD) with a learning rate set at 0.01 and momentum at 0.9. The momentum component aids the optimizer in moving towards the global minimum and Smooths out oscillations during updates. This model architecture is purposefully simple to assess the baseline effectiveness of logistic regression on image data.

Training Process: Learning from Data The model training involved adjusting the model weights to reduce the classification error on the training dataset. Important components of the training process included: • Epochs and Batching: The model underwent 10 complete passes (epochs) over the training data. To enhance the efficiency and scalability of the training process, data was processed in mini-batches of 64, which strikes a balance between memory use and stable gradient estimation. • Forward Pass: For each batch, the model calculated the class scores for each image. • Loss Computation: The discrepancy between predicted probabilities and actual labels was evaluated using cross-entropy loss. • Backward Pass: The model executed back propagation to determine the gradients of the loss in relation to the model weights. • Parameter Updates: The gradients were utilized by the SGD optimizer to modify the model's weights, steering it closer to a solution with minimal loss. During the training process, we tracked the training loss and accuracy at every epoch to assess the convergence.

progress of learning and Evaluation: Performance Measuring Model Conventional methods such as logistic regression still hold relevance in today's AI environment—particularly when performance was evaluated using the 10,000 images from the test set, which the model had not encountered previously. A blend of quantitative and visual assessment methods was employed: • Accuracy: This was determined as the ratio of accurately predicted digits to the total number of test samples. This metric provides a clear indication of the model's overall classification efficacy. • Confusion Matrix: A confusion matrix was created to analyse the model's misclassification trends. It illustrates the distribution of predicted classes in comparison to actual classes, revealing which digits were frequently mistaken for one another. • Loss and Accuracy Curves: Graphs were generated using Matplotlib and Seaborn to display how loss decreased and accuracy increased over the epochs. These visualizations assist in identifying training problems such as over fitting or under fitting;

IV CONCLUSION

Using logistic regression for handwritten digit recognition in PyTorch demonstrates that even the simple's machine learning models can be quite effective. Although intricate deep learning architectures like CNNs often attract attention for their superior accuracy, they typically require significant computational resources and can be more challenging to interpret. In contrast, logistic regression is straightforward, quick, and easy to grasp— making it ideal for scenarios where rapid outcomes, clear insights, or limited resources are critical. With the help of PyTorch's adaptable architecture and GPU capabilities, we were able to train the model efficiently and swiftly. Despite its simple configuration, the model delivered surprisingly strong results on the MNIST dataset. This illustrates that transparency and speed are just as important as achieving high accuracy.

References

1. Hazan H, Saunders DJ, Khan H, Patel D, Sanghavi DT, Siegelmann HT, Kozma R. BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python. Front Neuroinform. 2018 Dec 12;12:89. doi: 10.3389/fninf.2018.00089. PMID: 30631269; PMCID: PMC6315182.
2. Choi, Hangseo & Jeong, Jongpil & Lee, Chaegyu & Yun, Seokwoo & Bang, Kyunga & Byun, Jaebeom. (2023). Design and Implementation for BIC Code Recognition System of Containers using OCR and CRAFT in Smart Logistics. WSEAS TRANSACTIONS ON COMPUTER RESEARCH. 11. 62-72. 10.37394/232018.2023.11.6.
3. Liman, Muhamad & Josef, Antonio. (2024). Handwritten Character Recognition using Deep Learning Algorithm with Machine Learning Classifier. JOIV: International Journal on Informatics Visualization. 8. 10.62527/joiv.8.1.1707.
4. C. C. Tappert, C. Y. Suen and T. Wakahara, "The state of the art in online handwriting recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 8, pp. 787-808, Aug. 1990, doi: 10.1109/34.57669.
5. T. Akter, M. S. Akter, T. Mahmud, R. Chakma, M. S. Hossain and K. Andersson, "Evaluating the Performance of Machine Learning Models in Handwritten Signature Verification," 2024 Asia Pacific Conference on Innovation in Technology (APCIT), MYSORE, India, 2024, pp. 1-6, doi: 10.1109/APCIT62007.2024.10673648.
6. Anna Agius, Marie Morelato, Sébastien Moret, Scott Chadwick, Kylie Jones, Rochelle Epple, James Brown, Claude Roux,Dataset of coded handwriting features for use in statistical modelling,Data in Brief,Volume 16,2018,Pages 1010-1024,ISSN 2352 3409
7. Shah, Faisal & Kamran, Yousaf. (2007). Handwritten Digit Recognition Using Image Processing and Neural Networks. Lecture Notes in Engineering and Computer Science. 2165.
8. Millán-Hernández, Christian Eduardo & García Hernández, René Arnulfo & Ledeneva, Yulia. (2019). An evolutionary logistic regression method to identify confused drug names. Journal of Intelligent & Fuzzy Systems. 36. 1-11. 10.3233/JIFS-179012.
9. J. Li, G. Sun, L. Yi, Q. Cao, F. Liang and Y. Sun, "Handwritten Digit Recognition System Based on Convolutional Neural Network," 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), Dalian, China, 2020, pp. 739-742, doi: 10.1109/AEECA49918.2020.9213619.
10. Alejandro Baldominos, Yago Saez, Pedro Isasi,Evolutionary convolutional neural networks: An application to handwriting recognition,Neurocomputing,Volume 283,2018,Pages 38-52,ISSN 0925-231