# Handwritten Digit Recognition Using MNIST with Convolutional Neural Network

**Akanksha Naik**
Department of Computer Science and Engineering (AI&ML)
CMR Engineering College
208R1A6661@cmrec.ac.in

***Dr. M Kumara Swamy***
Department of Computer Science and Engineering (AI&ML)
CMR Engineering College
m.kumaraswamy@cmrec.ac.in

***Saubhik Ghosh***
Department of Computer Science and Engineering (AI&ML)
CMR Engineering College
208R1A66A4@cmrec.ac.in

***Sumit Sharma***
Department of Computer Science and Engineering (AI&ML)
CMR Engineering College
*208R1A66A8@cmrec.ac.in*

## Abstract

Handwritten digit recognition has long been a subject of interest in the fields of computer vision and pattern recognition. It used to be difficult to recognize handwritten numbers, but with the advancement of machine learning algorithms, this is no longer an issue. Convolutional neural networks (CNN) have made significant advances in machine learning in recent years to improve the recognition accuracy of different handwriting styles. This study develops a deep CNN model with a fast- convergence rate in training to significantly enhance the Modified National Institute of Standards and Technology (MNIST) database hand-written digit dataset recognition rate. The suggested model has a multi-layer deep arrangement structure that includes one fully connected layer (i.e., dense layers) for classification and two convolution and activation layers for feature extraction. To improve the recognition performance, the activation function, learning rate, batch sizes, kernel sizes, batch normalization, and other hyperparameters of the model are improved. The average accuracy of the proposed methodology was determined to be 99.07% on the test. It indicates a significant improvement in the proposed approach.

***Keywords:*** *MNIST Dataset, Deep Learning, Convolutional Neural Network, Handwritten Digit Recognition, Pattern Recognition, Computer Vision.*

## I. INTRODUCTION

The ability of machines to identify handwritten information created by humans is known as handwriting recognition. It is far more difficult for the machine to handle the range of handwriting styles, spacing differences, and handwriting errors. Continually, machine learning models are developing and have changed dramatically in the last few years.

Numerous cutting-edge models can attain very high precision and excellent performance. Since it was so successful, this technology is currently utilized for a variety of tasks, including data entry on forms, processing bank checks, and reading postal addresses.

Convolutional neural networks (CNNs) are widely and conveniently used for these image recognition and classification tasks. CNN is a special type of neural network capable

of taking in an input image, assigning importance to various aspects, and being able to distinguish one from another. Recent research has seen convolutional neural networks being applied for facial recognition, document analysis, speech detection, and license plate recognition [1].

The convolution operation is used to build and train many models; however, obtaining high accuracy depends on several elements, including the network design and dataset used. Using the same dataset—the Modified National Institute of Standards and Technology (MNIST) dataset—we experimented to determine how alternative topologies could impact the accuracy of handwritten digit recognition. MNIST is a massive handwritten-digit database with 60,000 grayscale pictures, each measuring 28 x 28 pixels. There are ten classes total, representing the numerals 0 through 9. The digit images are centered and standardized in size, making this a great dataset for assessment.

We have several reasons for using this MNIST handwritten-digit database. First of all, it is a standard and reasonably fundamental database for quickly testing ideas and algorithms, as was previously described. The handwritten digits in the MNIST database have already undergone segmentation and normalization as part of the preprocessing that we want to test neural networks applied to real-world practical problems. We can compare our results with those from a rather extensive body of literature because many academics use MNIST to evaluate their theories and techniques.

Artificial neural networks, or ANNs, are known to be inspired by the biological central systems found in brains. They have been applied to a wide range of tasks, including multivariate data processing, modeling, classification, and pattern recognition [3, 20]. Furthermore, back-propagation neural networks are frequently used in real-world ANN applications. In 1986, D. Rumelhart and J. McCelland proposed the back-propagation algorithm [21]. They also introduced backpropagation neural networks (BPNNs), which are neural networks that use BP techniques [21]. To solve the classification problem, we employ back-propagation neural networks in this research. Supervised multi-layer feed-forward neural networks, or backpropagation neural networks, typically have an input layer, an output layer, and one or more hidden layers [3].

## II. LITERATURE REVIEW

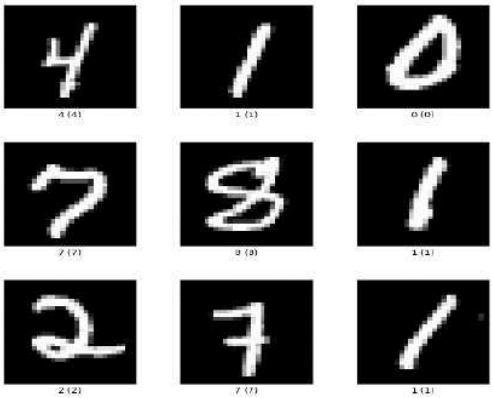

**Figure 1:** MNIST Digits Classification

**Similar Projects**

Handwriting recognition has already achieved impressive results using shallow networks [9, 10, 11, 12, 13, 14, 15, 16, 17]. Many papers have been published with research detailing new techniques for the classification of handwritten numerals, characters, and words. The deep belief networks (DBN) with three layers along with a greedy algorithm were investigated for the MNIST dataset and reported an accuracy of 98.75% [4]. Pham et al. applied a regularization method of dropout to improve the performance of recurrent neural networks (RNNs) in recognizing unconstrained handwriting [5]. The author reported improvement in RNN performance with a significant reduction in the character error rate (CER) and word error rate (WER).

In 2003, Simard et al. introduced a general convolutional neural network architecture for visual document analysis and weeded out the complex method of neural network training [6].

The performance of CNNs depends mainly on the choice of hyper-parameters [7], which are usually decided on a trial-and-error basis. Some of the hyper-parameters are, namely, activation function, number of epochs, kernel size, learning rate, hidden units, hidden layers, etc. These parameters are very important, as they control the way an algorithm learns from data [8]. Hyper-parameters differ from model parameters and must be decided before the training begins.

Basri et al. [18] observed and compared the performance contributed by different networks. The four models being discussed in the paper are AlexNet, MobileNet, GoogLeNet, and CapsuleNet. While considering the result from normal data in the same condition, the error rate of relevant models is GoogLeNet (Inception V3) at 7%, AlexNet at 8%, CapsuleNet at 8.7%, and MobileNet at 20.3%. After adding the augmentation dataset to the training process, the error rate was reduced to AlexNet at 0.99%, GoogLeNet at 1.49%, CapsuleNet at 7.76%, and MobileNet at 16.42%. For the computation time, AlexNet was fastest at 1.14s, CapsuleNet at 3.86s, MobileNet at 12.52s, and GoogLeNet at 22.53s. This implies that the structure of models results in different performance and computation times, and it also emphasizes the importance of augmented datasets.

## III. METHODOLOGY AND APPROACH

The dataset used in this paper is the MNIST dataset for handwritten digits. This is a dataset of 60,000 28x28 grayscale images of the 10 digits, along with a test set of 10,000 images. There are ten classes total, representing the numerals 0 through 9. Yann LeCun and Corinna Cortes generated and assembled the initial work in 1999, from which the database is derived. The digit images are centered and standardized in size, making this a great dataset for assessment. With the help of the labels, the training dataset teaches the model what each digit looks like. The model is then tested using the test dataset, which consists solely of photos, to determine if it can predict data that it has never seen before.

- **Convolutional Neural Network:** Artificial neural networks and the most modern deep learning techniques are combined in convolutional neural networks. They have been applied for many years to image identification problems, such as the recognition of handwritten digits, which is the topic of this work [1]. A regularized kind of feed-forward neural network, the convolutional neural network (CNN) uses filter (or kernel) optimization to teach itself feature engineering. By applying regularized weights over

fewer connections, backpropagation issues with disappearing gradients and expanding gradients—which were observed in previous neural networks—are avoided.

- **Keras:** Keras is a deep-learning Python API capable of running on top of either JAX, TensorFlow, or PyTorch. It's not overly simplistic, but simple. By lowering the cognitive load on developers, Keras allows them to concentrate on the crucial aspects of the problem. Keras adheres to the progressive disclosure of complexity concept, which states that while arbitrarily advanced workflows should be achievable through a clear path that builds upon prior knowledge, simple workflows should be rapid and simple. Keras offers scalability and performance that rivals that of the industry; NASA, YouTube, and Waymo are just a few of the companies that employ it [19].

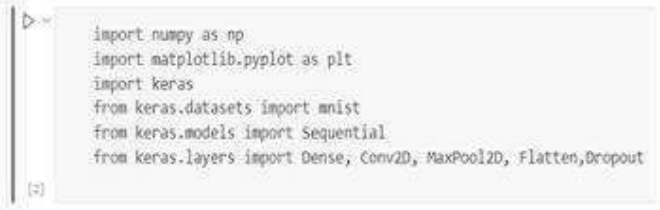


**Figure 2:** Importing Libraries

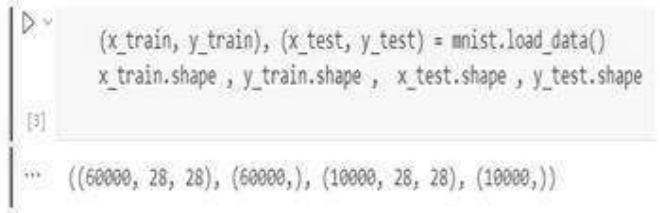


**Figure 3:** Loading Of Dataset

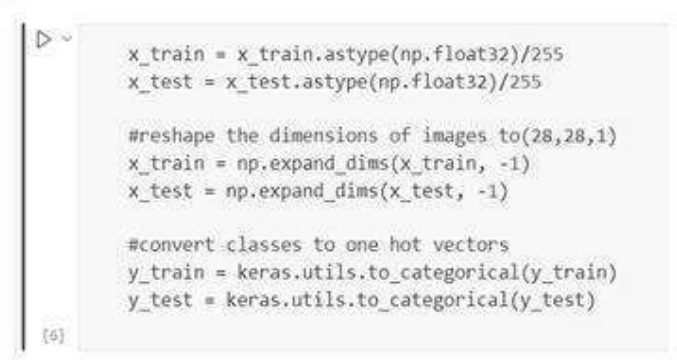


**Figure 4:** Data Transformation

- **Recommendation:** To put it briefly, CNN is frequently utilized for picture classification issues. More innovative and effective CNN architectures are created as a result of the complexity of datasets and the constant progress in technology. We must select the appropriate architecture for the given challenge, spoiled by the multitude of CNN architecture options.

Additionally, while we worked on the project, we discovered that the model created through training might vary depending on even the smallest modifications to the architecture or configuration of the device, and it may not even perform the same way when running on the same system. For this reason, you should exercise caution when choosing the architecture.

- **Algorithm Implementation:** The database that we have employed in the project is the MNIST database. The database is imported using the Keras deep learning API and is sourced from the original work, which was created and compiled by Yann LeCun and Corinna Cortes in 1999. It contains 70,000 grayscale images, which are split into train and test sets of 60,000 and 10,000 images, respectively. From numbers 0 to 9, there are a ton of grayscale photographs in both collections. While the photos in the testing dataset lack labels, the images in the training dataset have labels corresponding to their classes.[1]


**Figure 5:** CNN Model Architecture

a. **Preparation:** To ensure that everything works properly, there are a few things that need to be done before the session begins. Before using the functions further, the developer must first set up the environment by importing the necessary libraries. Then the MNIST dataset is loaded using the Keras API. Then the data from the dataset is transformed to meet the needs of our project by reshaping and converting the classes into one hot vector.
b. **Training:** CNN is the architecture and algorithm in use. The model will receive the image data as input and produce the predicted result. As a result, learning occurs when the actual outcome confirms the expected result and provides input to the model for improvement.
c. **Evaluation:** We evaluate the model after it has been trained on the developer's specified architecture configuration. To do this, we compute the accuracy and loss as well as the validation accuracy and loss. We then plot the graph and obtain the training and testing accuracy using those metrics.


**Figure 6:** Evaluation Scores

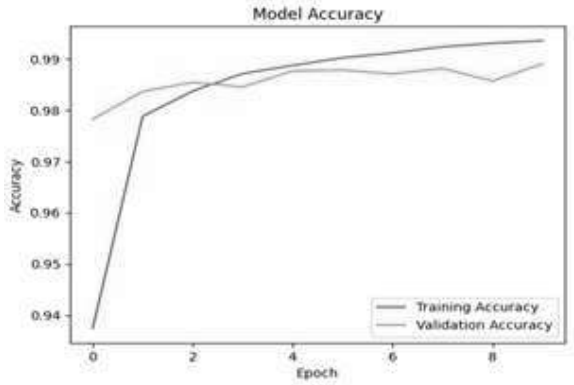


**Figure 7:** Training and Validation Graph

## IV. RESULTS AND DISCUSSION

- **Discussion on Implementation:** Proposing a faster and more accurate architecture to solve the MNIST handwritten digit image classification problem is the goal. We have experimented with different CNN algorithm setups to determine which one gets us closest to our objective of quick and accurate recognition. After integrating two convolution layers and two max-pooling layers into a dense layer and running ten epochs, we were able to obtain the a.h5 model file.

- **Model Optimization and Performance Evaluation:** By specifying the cross-entropy loss function and automatically minimizing the loss function during training with the Adam optimizer, to improve the network's fit to the training sample data, the loss will be back propagated during this procedure. The dropout rate is set to 0.25. After ten iterations, it is evident that the training loss and accuracy converge, as seen in Table 1 and Figure 7.

**Table 1:** Model Training Loss and Accuracy

| Epoch | Time | Loss | Accuracy | Validation Loss | Validation Accuracy |
|---|---|---|---|---|---|
| 1 | 14s | 0.2052 | 0.9374 | 0.0742 | 0.9782 |
| 2 | 13s | 0.0681 | 0.9788 | 0.0559 | 0.9836 |
| 3 | 13s | 0.0522 | 0.9837 | 0.0477 | 0.9854 |
| 4 | 13s | 0.0420 | 0.9871 | 0.0483 | 0.9845 |
| 5 | 13s | 0.0352 | 0.9887 | 0.0422 | 0.9876 |
| 6 | 13s | 0.0300 | 0.9902 | 0.0390 | 0.9878 |
| 7 | 13s | 0.0275 | 0.9911 | 0.0447 | 0.9871 |
| 8 | 13s | 0.0243 | 0.9923 | 0.0386 | 0.9881 |
| 9 | 13s | 0.0214 | 0.9930 | 0.0525 | 0.9857 |
| 10 | 13s | 0.0190 | 0.9935 | 0.0426 | 0.9890 |

## V. CONCLUSION

With the development of increasingly complex neural networks by researchers, the task of image recognition continues to expand and evolve. Research revealed that the

performance of models will vary depending on the task being completed. We consider a variety of parameters when evaluating a model's fitness for a certain task, not only accuracy and error rate. Along with accuracy and error rate, training time is a significant element; the more complicated and larger a dataset is, the more critical training time is. Additionally, we discovered a few shortcomings when trying to manually draw the digits. These can vary from user to user since each person writes differently, therefore it seems clear that there is room for improvement and that the accuracy is not 100% accurate yet.

## References


[1] MNIST handwritten digit recognition with different CNN architectures – Lead Ming Seng, Brennan Bang Chen Chiang, Zailan Arabee Abdul Salam, Gwo Yih Tan, Hui Tong Chai. *Journal of Applied Technology and Innovation (e -ISSN: 2600-7304) vol. 5, no. 1, (2021)*

[2] MNIST Handwritten Digit Classification Based on Convolutional Neural Network with Hyperparameter Optimization – Haijian Shao, Edwin Ma, Ming Zhu, Xing Deng, Shengjie Zhai. *Intelligent Automation & Soft Computing 36(3):3595 DOI:10.32604/iasc.2023.036323, (2023)*

[3] Classification of MNIST Handwritten Digit Database using Neural Network – Wan Zhu. *Research School of Computer Science, Australian National University, Acton, ACT 2601, Australia, (2012)*

[4] Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast-learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554.

[5] Pham, V.; Bluche, T.; Kermorvant, C.; Louradour, J. Dropout improves recurrent neural networks for handwriting recognition. In Proceedings of the 14th Int. Conf. on Frontiers in Handwriting Recognition, Heraklion, Greece, 1–4 September 2014.

[6] Simard, P.Y.; Steinkraus, D.; Platt, J.C. Best practice for convolutional neural networks applied to visual document analysis. In Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), Edinburgh, UK, 3–6 August 2003.

[7] Cui, H.; Bai, J. A new hyperparameters optimization method for convolutional neural networks. *Pattern Recognition. Lett.* **2019**, *125*, 828–834.

[8] Tso, W.W.; Burnak, B.; Pistikopoulos, E.N. HY-POP: Hyperparameter optimization of machine learning models through parametric programming. *Comput. Chem. Eng.* **2020**, *139*, 106902.

[9] Choudhary, A.; Ahlawat, S.; Rishi, R. A neural approach to cursive handwritten character recognition using features extracted from binarization technique. *Complex Syst. Model. Control Intell. Soft Comput.* **2015**, *319*, 745–771.

[10] Choudhary, A.; Rishi, R.; Ahlawat, S. Handwritten numeral recognition using modified BP ANN structure. In Proceedings of the Communication in Computer and Information Sciences (CCIS-133), Advanced Computing, CCSIT 2011, Royal Orchid Central, Bangalore, India, 2–4 January 2011; Springer: Berling/Heildelberg, Germany, 2011; pp. 56–65.

[11] Cai, Z.W.; Li-Hong, H. Finite-time synchronization by switching state-feedback control for discontinuous Cohen–Grossberg neural networks with mixed delays. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1683–1695.

[12] Zeng, D.; Dai, Y.; Li, F.; Sherratt, R.S.; Wang, J. Adversarial learning for distantly supervised relation extraction. *Comput. Mater. Contin.* **2018**, *55*, 121–136.

[13] Long, M.; Yan, Z. Detecting iris liveness with batch normalized convolutional neural network. *Comput. Mater. Contin.* **2019**, *58*, 493–504.

[14] Chuangxia, H.; Liu, B. New studies on dynamic analysis of inertial neural networks involving non-reduced order method. *Neurocomputing* **2019**, *325*, 283–287.

[15] Xiang, L.; Li, Y.; Hao, W.; Yang, P.; Shen, X. Reversible natural language watermarking using synonym substitution and arithmetic coding. *Comput. Mater. Contin.* **2018**, *55*, 541–559.

[16] Huang, Y.S.; Wang, Z.Y. Decentralized adaptive fuzzy control for a class of large-scale MIMO nonlinear systems with strong interconnection and its application to automated highway systems. *Inf. Sci.* **2014**, *274*, 210–224.

[17] Choudhary, A.; Rishi, R. Improving the character recognition efficiency of feed-forward bp neural network. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *3*, 85–96.

[18] Basri, R. & Akter, M. (2020). Bangla Handwritten Digit Recognition Using Deep Convolutional Neural Network | Proceedings of the International Conference on Computing Advancements. [Online]. 2020. Doi.org. Available at https://doi.org/10.1145/3377049.3377077

[19] https://keras.io/api/datasets/mnist/ - Keras MNIST Dataset

[20] Basheer, I.A., and Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of microbiological methods, 43(1), pp.3-31.

[21] Rumelhart, D.E., Hinton, G.E. and McClelland, J.L., 1986. A general framework for parallel distributed processing. Parallel distributed processing: Explorations in the microstructure of cognition, 1, pp.45-76.
[22] Yann.lecun.com., n.d. MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. [online] Available at: http://yann.lecun.com/exdb/mnist/
[23] Yann.lecun.com., n.d. MNIST Demos on Yann LeCun's website. [online] Available at: http://yann.lecun.com/exdb/lenet/