International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

# Hierarchical Approaches to Handwritten Digit Recognition: A Study of Modern Neural Networks

**Ismail Hossain Sadhin[1], Elora Majumder Bandhan[2], Md. Abdullah Al Mamun[3]**

[1]Dept. of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh
[2]Dept. of Electrical and Electronics Engineering, American International University-Bangladesh Dhaka, Bangladesh
[3]Dept. of Electrical and Electronics Engineering, Chittagong University of Engineering & Technology Chittagong, Bangladesh

## Abstract

Handwritten digit recognition has several applications in multiple industries in this modern era for enhancing efficiency, accuracy, and accessibility. Convolution Neural Networking (CNNs) have emerged for the precise result of this task since it is a powerful tool for this task due to the ability to learn hierarchical features from data. Several architectures of convolution neural networking (CNNs) are used in this field. This study investigated the efficiency of CNN architectures in recognizing handwritten digits. The considered architectures are VGG16, ResNet50, and DenseNet. However, DenseNet stands out with the highest efficiency of 99.19% compared to traditional architectures like VGG16 and ResNet50. A benchmarked dataset of the machine learning field, MNIST, has been used for training data. Through experimental evaluation, it was observed that the three CNN architectures mentioned achieved high accuracy rates on the MNIST dataset, namely 95.9% for VGG16, 98.5% for ResNet50, and 99.19% for DenseNet. However, DenseNet has been proven to be the most accurate.

**Keywords:** Handwritten recognition, Digit recognition, MNIST, Neural network, Machine learning.

## 1. Introduction

AI-powered digit recognition converts handwritten digit images into digital digits that enhance efficiency, reduce time consumption, and provide accuracy. It plays a crucial role in multiple industries of various applications such as bank automation or mobile banking, postal office code number recognition, automation in depositing paper checks, Identity number recognition, automatic license plate recognition, and so on [1][2]. These real-life applications help us to make human life easier by solving complex problems. Therefore, developers using machine learning or deep learning techniques develop more intelligent solutions. Numerous algorithms have been developed for handwritten digit recognition. Among the techniques, Convolutional Neural Networking (CNN) from deep learning is significantly advanced and could provide mostly precise findings in the image identification field. It is used in various identification, classification, or recognition tasks, especially in image recognition and digitalization.

Convolution Neural Networks (CNNs) are mainly used for feature extraction or finding patterns for recognition. The architecture of CNN consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers [3]. The convolutional layer applies kernels to input data. After each convolutional layer, pooling layers typically follow to reduce the spatial size of the representation, making the network more computationally efficient and reducing the risk of overfitting by summarizing the presence of features in sub-regions. Toward the end of the network, fully connected layers are used to classify the extracted features into specific categories.

## 2. Related Work

Several studies have been conducted on handwritten digit recognition for its advancement. These studies have used various machine learning and deep learning techniques, as well as datasheets such as MNIST, SVHN, and USPS, to explore methodologies from traditional algorithms to conversant convolutional neural networks (CNNs) and to improve the accuracy and efficiency of recognition. The key areas of applications of digit recognition tasks are document processing, postal automation, biometric authentication, and so on.

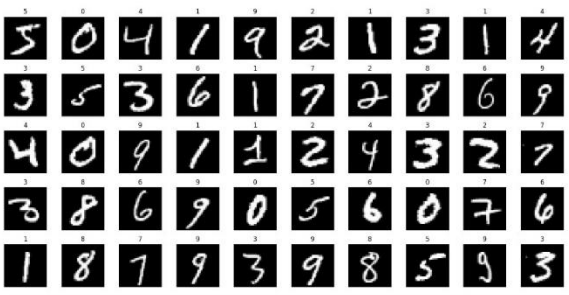
A study has explored the development of English handwritten recognition using deep neural networks (DNNs) architecture such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to extract meaningful features from handwritten samples and improve classification accuracy, focusing on large-scale datasets to overcome the challenges of noise in input data and variability in handwriting styles [4]. The study results show notable advancements in handwritten English character recognition that could be implemented in automated form processing, document digitalization, and better human-computer interactions [4].

Another inclusive study on handwritten digit recognition using machine learning algorithms was conducted to compare the efficiency of various methodologies in accurately classifying digits from handwritten images. It covered a range of traditional machine learning algorithms such as Support Vector Machines (SVM), k-nearest Neighbors (k-NN), Decision Trees, and ensemble methods like Random Forests [5].

## 3. Selected Handwritten Image Database

The database used for this research is a digit database from the MNIST database of handwritten digits. This database consists of 30,000 patterns from about 250 different writers in total and contains 60000 examples of the training set and 10000 examples of the test set. The digits are all uniform in size and centered in the 28x28 image [4]. The MNIST database is widely used for this kind of digit recognition system implementation. The sample of a training image for MNIST is shown in Figure 1.

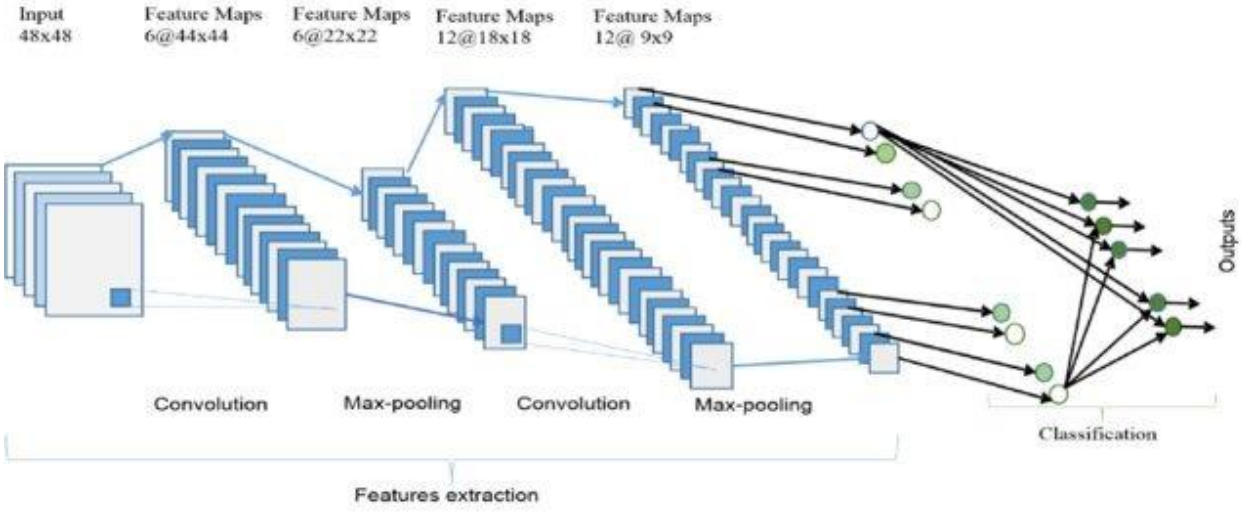
**Figure 1: Sample of MNIST Database**



## 4. Convolutional Neural Network (CNN) Architecture

Convolutional neural network is the most widely used deep learning model in feature learning for large-scale image classification and recognition. In several crucial areas, image processing or recognition can be done more precisely using a convolutional neural network (CNN). Among several vital fields of application, computerized tomography (CT) scan can be an example implemented with 3D CNN architecture, which is better than x-rays, and various diseases can be diagnosed early [6]. A convolutional neural network consists of three layers: the convolutional layer, the subsampling layer (pooling layer), and the connected layer [7].

**Figure 2: CNN Architecture [8]**




The CNN architecture consists of several layers or multi-building blocks—each layer in the CNN architecture, including its function. In CNN architecture, the most significant component is the convolutional layer. It consists of a collection of convolutional filters as kernels. The input image, expressed as N-dimensional metrics, is convolved with these filters to generate the output feature map. A grid of discrete numbers or values describes the Kernel. Each value is called the kernel weight. Random numbers are assigned to act as the weights of the Kernel at the beginning of the CNN training process. In Convolution layer operation, the vector format is the input of the traditional neural network, while the multi-channeled image is the input of the CNN. For instance, single-channel is the format of the gray-scale image, while the RGB image format is three-channeled. To understand the convolutional operation, let us take an example of a 4 X 4 gray-scale image with a 2 X 2 random weight-initialized kernel—first, the Kernel slides over the whole image horizontally and vertically. In addition, the dot product between the input image and the Kernel is determined, where their corresponding values are multiplied and then summed up to create a single scalar value, calculated concurrently. The whole process is then repeated until no further sliding is possible. Note that the calculated dot product values represent the feature map of the output. Figure 8 graphically illustrates the primary calculations executed at each step. In this figure, the light green color represents the 2 X 2 kernel, while the light blue color represents an area of similar size to the input image. After summing up the resulting product values (marked in a light orange color), both are multiplied; the result represents an entry value to the output feature map. However, padding to the input image is not applied in the previous example, while a stride of one is applied to the Kernel [9]. It can be formulated with- [i-k]+1, where i is the input size and k is the size of the Kernel. Stride is a

parameter of the neural network's filter that modifies the amount of movement over the image or video. We had stride 1, so it will take one by one. If we give stride 2, it will take value by skipping the next 2 pixels. This can be formulated- [i-k/s]+1.

Padding is a term relevant to convolutional neural networks as it refers to the number of pixels added to an image when the Kernel of a CNN is processing it. For example, if the padding in a CNN is set to zero, then every added pixel value will be of value zero. When we use the filter or Kernel to scan the image, the size of the image will go smaller. We have to avoid that because the original size of the image needs to be preserved to extract some low-level features. Therefore, we will add some extra pixels outside the image. For padding, the formula is [i-k+2p/s]+1

Pooling in convolutional neural networks is a technique for generalizing features extracted by convolutional filters and helping the network recognize features independent of their location in the image. Finally, flattening converts all the resultant 2-dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image [10].

## 5. Methodology

### 5.1. VGG16

The convolutional neural network architecture known as VGG16, abbreviated as the VGG16-layer model, is well-known for its simplicity and usefulness in image recognition applications. It has 16 layers, 13 convolutional and three fully connected. The vital quality of VGG16 is its uniform engineering, where a maximum pooling layer trails each convolutional layer to lessen spatial aspects. The convolutional layers use tiny 3x3 filters with a stride of one, which keeps the receptive field small and makes it possible to learn more about representations [11]. This planning system, combined with its profound heap of layers, permits VGG16 to gain multifaceted highlights from input pictures, making it especially capable of perceiving objects in different visual settings. VGG16 is a popular choice for transfer learning and as a baseline model in computer vision research and applications due to its straightforward architecture despite its depth.

### 5.2. ResNet50

ResNet50 is a profound convolutional brain network design that upset the field of PC vision by presenting lingering associations, tending to the test of preparing profound organizations successfully. ResNet50 is a member of the ResNet family and is notable for its depth and performance. Researchers at Microsoft developed it. The architecture has 50 layers, each stage containing multiple residual blocks, and is divided into stages. These blocks make it possible for the network to learn residual mappings, in which each block learns the residual function concerning the input, making it easier to train intense networks. A shortcut connection that adds the original input to the output is followed by two or three convolutional layers with batch normalization and ReLU activations in each residual block [12]. This skip association mitigates the corruption issue by permitting slopes to stream all the more straightforwardly during backpropagation, subsequently empowering proficient preparation and working on the general exactness of the model. ResNet50's design has demonstrated compelling across a scope of PC vision errands, including picture grouping, object recognition, and division, accomplishing cutting-edge results on different benchmarks like ImageNet. Due to its modular design and transfer learning capabilities, it is a popular choice for research and practical applications that require robust and high-performance deep learning models.

### 5.3. DenseNet

DenseNet, abbreviated as Dense Convolutional Network, is a novel architecture for a convolutional neural network created to address parameter efficiency issues and feature propagation. DenseNet, developed by Facebook AI Research (FAIR) researchers, stands out due to its dense layer connectivity, in which each layer receives input from all layers before it and transmits its feature maps to all layers below it. Compared to more conventional architectures like VGG or ResNet, this dense connectivity encourages feature reuse throughout the network. A "dense block" is created in DenseNet when the output of each layer is combined with the feature maps of all of the layers that came before it. Multiple convolutional layers are typically used within each dense block to improve feature learning and propagation, followed by batch normalization and ReLU activations [13]. The thick associations guarantee that the inclination stream is amplified during preparing, advancing component reuse, and reinforcing highlight proliferation across the organization.

Additionally, this dense connectivity encourages feature diversity and alleviates the vanishing gradient issue, improving model performance. DenseNet architectures are distinguished by their efficiency and compactness, enabling them to achieve cutting-edge results on various image classification benchmarks with fewer parameters. The approach's adaptability and robustness across various computer vision applications have been demonstrated by its application to image segmentation and object detection. DenseNet is a compelling option for deep learning research and practical applications due to its distinctive dense connectivity structure and parameter efficiency advantages.

### 5.4. MLP Classifier

A multilayer perceptron (MLP) classifier may be a feedforward manufactured neural organization comprising numerous layers of hubs, counting an input layer, one or more covered-up layers, and an output layer. Each hub within the organization may be a perceptron that forms data and passes it to hubs within the following layer. In an MLP classifier, each layer but the input layer contains neurons that utilize activation functions to convert weighted inputs from the past layer into yields. The covered-up layers empower the organization to memorize complex designs within the input information, whereas the yield layer produces the ultimate classification or expectation. MLP classifiers are prepared utilizing administered learning strategies such as backpropagation, where the arrange alters its weights based on the mistake between anticipated and genuine yields, pointing to play down this blunder over time. This adaptability and capacity to show nonlinear connections make MLP classifiers capable instruments for errands like picture acknowledgment, standard dialect preparation, and other complex design acknowledgment issues [14].

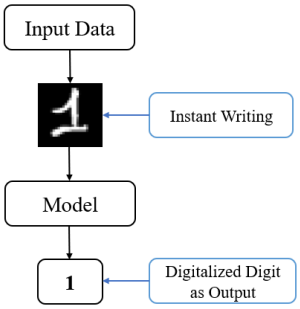
### 5.5. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a robust supervised machine learning algorithm for classification and regression tasks. Its primary objective in classification is to find the optimal hyperplane that best separates data points of different classes in a high-dimensional space. The hyperplane is chosen to maximize the margin between the closest data points from each class, known as support vectors. SVMs effectively handle linear and nonlinearly separable datasets through kernel functions, which map input data into higher-dimensional feature spaces where separation is more straightforward. SVMs are remarkably robust against overfitting, aiming to maximize the margin, which helps generalize well to unseen data [15]. SVMs are widely used in various domains, including text categorization, image recognition, and bioinformatics, owing to their ability to handle complex decision boundaries and perform well on small to medium-sized datasets.

## 6. Implementation

In this study, we implemented a handwritten digit recognition machine using the different algorithms of Deep learning and machine learning. Since it is an image recognition method, a convolution neural network (CNN) works precisely in this field. However, a model has been implemented and verified to produce the highest efficiency.

**Figure 3: Implementation of the study**




In Figure 3, the implementation steps are shown. Firstly, data is given as input, for example, a handwritten one (1), and then it is processed through the model. This model is made with several deep learning algorithms such as VGG16, ResNet50, and DenseNet and machine learning algorithms such as MLP classifier and SVM. After processing through the model, the result is digitalized 1. Among these algorithms, DenseNet has the highest efficiency with 99.19%. So, DenseNet is the most precise algorithm for handwritten digit recognition.

## 7. Result & Analysis

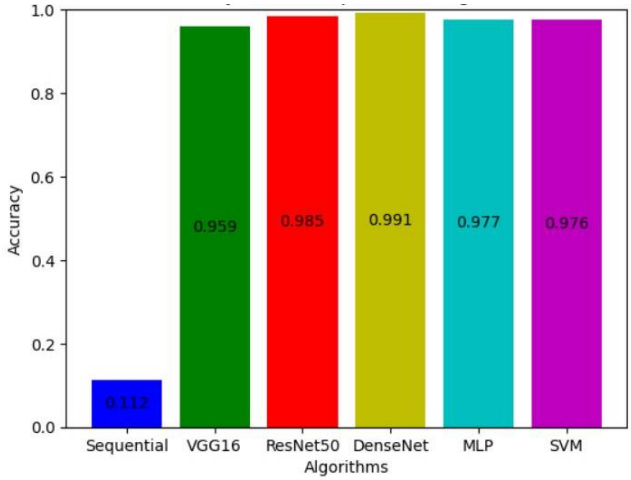
This study employed an inclusive approach to evaluate six distinct algorithms from machine learning and deep learning domains to achieve optimal performance in the Handwritten Digit recognition field. It was conducted to get a result from each algorithm regarding accuracy, losses, and execution time to state the algorithm as most accurate and efficient for the task. Initially, we applied the Sequential Model, a foundational neural network architecture that provided a baseline for performance. Next, we explored VGG16, a convolutional neural network that came up with improved feature extraction and recognition accuracy. After that, ResNet50 is employed to enhance model performance by enabling the training of much deeper networks. DenseNet was another critical component of our analysis, as its dense connectivity pattern facilitates improved gradient flow and feature reuse, leading to more robust and efficient learning. Further, we incorporated machine learning algorithms such as Multilayer Perceptron (MLP), which can effectively capture patterns in data and Support Vector Machines (SVM), a powerful technique in classification, especially in scenarios with margins of differentiation.

After implementing these algorithms, the accuracy and losses were compared to identify the most effective approach for recognizing handwritten digits. The outcome and analysis have been compared and visualized below with a table and graphical representation. Those assist in distinguishing the most effective algorithm for handwritten digit recognition. The analysis is illustrated below-

**Table 1: Accuracy and Losses for Different Algorithms**

| Name of Algorithms | Test Accuracy level | Losses |
|---|---|---|
| **Sequential** | 11.2% | 88.8% |
| **VGG16** | 95.9% | 4.1% |
| **ResNet50** | 98.5% | 1.5% |
| **DenseNet** | 99.1% | 0.9% |
| **MLP** | 97.7% | 2.3% |
| **SVM** | 97.6% | 2.4% |

**Figure 4: Accuracy Level Comparison of Algorithms**




In Figure 2, the graphical representation shows where the accuracy levels indicated by each algorithm. Among those algorithms, the lowest accuracy level is 11.2% which is for Sequential and the highest accuracy is 99.1% which could be extracted from the DenseNet algorithm. Based on the accuracy level, we declare DenseNet, as the most effective and efficient algorithm for the Handwritten Digit Recognition task.

## 8. Conclusion and Future Scope

This study investigated the most effective algorithm in the Handwritten digit recognition field. To study, several distinguished algorithms have been tested in the Machine learning and Deep learning domain using the MNIST database. After testing the algorithms, DenseNet provided the optimal performance based on accuracy and loss parameters. The accuracy level is 99.1% for Handwritten Digit Recognition using the DenseNet algorithm. This study is a preliminary effort to find an effective algorithm and make handwritten text easier to recognize. In the future, according to this study, handwritten digit recognition is expected to achieve similar or higher accuracy, even in complex and noisy databases or in real-time processing. This could significantly improve various applications of this method, such as real-time data entry, postal services, banking education, and a few more applications.

## 9. References

1. Subasi, “Other classification examples,” in Elsevier eBooks, 2020, pp. 323–390. doi: 10.1016/b978-0-12-821379-7.00005-9

2. M. Z. Alom, P. Sidike, M. Hasan, T. M. Taha, and V. K. Asari, "Handwritten Bangla Character Recognition Using the State-of-the-Art Deep Convolutional Neural Networks," Computational Intelligence and Neuroscience, vol. 2018, pp. 1–13, Aug. 2018, doi: 10.1155/2018/6747098.
3. A. Hidaka and T. Kurita, "Consecutive Dimensionality Reduction by Canonical Correlation Analysis for Visualization of Convolutional Neural Networks," Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and Its Applications, vol. 2017, no. 0, pp. 160–167, Jan. 2017, doi: 10.5687/sss.2017.160.
4. T. S. Gunawan, A. F. R. M. Noor, and M. Kartiwi, "Development of English Handwritten Recognition Using Deep Neural Network," Indonesian Journal of Electrical Engineering and Computer Science, vol. 10, no. 2, p. 562, May 2018, doi: 10.11591/ijeecs.v10.i2.pp562-568
5. S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, and A. A. Jobair, "Handwritten Digit Recognition Using Machine Learning Algorithms," Indonesian Journal of Science and Technology, vol. 3, no. 1, p. 29, Apr. 2018, doi: 10.17509/ijost.v3i1.10795.
6. Md. I. H. Sadhin, M. F. Woishe, N. Sultana, and T. Z. Bristy, "Identifying Lung Cancer Using CT Scan Images Based On Artificial Intelligence," Sadhin | International Journal of Computer and Information System (IJCIS), Mar. 18, 2022. https://www.ijcis.net/index.php/ijcis/article/view/64/65
7. A. D. Torres, H. Yan, A. H. Aboutalebi, A. Das, L. Duan, and P. Rad, "Patient Facial Emotion Recognition and Sentiment Analysis Using Secure Cloud With Hardware Acceleration," in Elsevier eBooks, 2018, pp. 61–89. doi: 10.1016/b978-0-12-813314-9.00003-7.
8. M. Z. Alom et al., "A State-of-the-Art Survey on Deep Learning Theory and Architectures," Electronics, vol. 8, no. 3, p. 292, Mar. 2019, doi: 10.3390/electronics8030292.
9. L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, vol. 8, no. 1, Mar. 2021, doi: 10.1186/s40537-021-00444-8.
10. Dharmaraj, "Convolutional Neural Networks (CNN) — Architecture Explained," Medium, Sep. 27, 2022. [Online]. Available: https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243
11. S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," International Journal of Scientific and Research Publications, vol. 9, no. 10, p. p9420, Oct. 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.
12. S. Poudel, Y. J. Kim, D. M. Vo, and S.-W. Lee, "Colorectal Disease Classification Using Efficiently Scaled Dilation in Convolutional Neural Network," IEEE Access, vol. 8, pp. 99227–99238, Jan. 2020, doi: 10.1109/access.2020.2996770
13. Gao & Liu, Zhuang & Weinberger, and Kilian, "Densely Connected Convolutional Networks," arXiv:1608.06993v5 [cs.CV] 28 Jan 2018, no. 12, Aug. 2026, [Online]. Available: https://www.researchgate.net/publication/319770123_Densely_Connected_Convolutional_Networks
14. Z. Chi, "MLP classifiers: overtraining and solutions," Nov. 2002, doi: 10.1109/icnn.1995.488180.
15. Farid, Nahla & Elbagoury, Bassant & Roushdy, Mohamed & M.Salem, and Abdel-Badeeh, "A Comparative Analysis for Support Vector Machines for Stroke Patients," 2013. https://www.researchgate.net/publication/255822126_A_Comparative_Analysis_for_Support_Vector_Machines_for_Stroke (accessed Aug. 30, 2024).