Optical Character Recognition System for Digit Recognition Using Deep Learning

Mona Awad[1], Marwa M. Eid[2]

[1] Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt
[2]Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 11152, Egypt
Emails: Dr.mona711@gmail.com; mmm@ieee.org

Abstract

Because it is so difficult to distinguish handwritten digits, digit identification is one of the most critical applications in computer vision. This is one of the reasons why it is so tough. The field of handwritten character recognition is one in which a great deal of application of numerous deep learning models has occurred. The startling parallels that can be drawn between deep learning and the brain are primarily responsible for its meteoric rise in popularity. In this study, the Artificial Neural Network and the Convolutional Neural Network, two of the most used Deep Learning algorithms, were investigated with an eye toward the recognition process's feature extraction and classification phases. With the assistance of the categorical cross-entropy loss and the ADAM optimizer, the models were trained on the MNIST dataset. Backpropagation and gradient descent are the two methods utilized during the training process of neural networks that contain reLU activations and carry out automatic feature extraction. In computer vision, one of the most common and widely used classifiers is the Convolution Neural Network, sometimes referred to as ConvNets or Convolutional neural networks. This network is used for the recognition and categorization of images.

Keywords: OCR; AI; Neural Network.

1. Introduction

When we hear the phrase "Handwritten Character Recognition," our minds immediately go to OCR, which stands for "optical character recognition." This technique reads characters and translates them into a structure that a machine can interpret. For the past three to four decades, this is the sole method that academics have used to transform any paper record into a digital version that a computer can alter. The first step in the OCR pipeline is picture acquisition, pre-processing, and segmentation. Finally, the pipeline concludes with decoding. Feature of the Data After the data have been segmented and extracted, the next step is to classify it. In a manner analogous to that of a pipeline, OCR calculates the rate of progression for each subsequent stage based on the proportion of successful completions of earlier stages. Because of how far technology has come, even the most challenging jobs can now be done entirely by machine. There are many different uses for computer vision, including report interpretation. The second choice is language translation, the third is organizing the mail, the fourth is reviewing the payments; etc.

As a consequence of this, numerous industry professionals are presently displaying a significant amount of interest in OCR. The success of the identification system is strongly dependent on the classification methods used. This is because written material can be found in many languages and subject areas. Consequently, it was a very difficult task to make this kind of structure more practical. Before we entered the age of artificial intelligence, many doors were opened by AI. To develop a classifier, one needs to manually extract features and then put their faith in the computer's ability to decide based on the data it has been given. In the past, successful algorithms such as SVM required a significant amount of labor and mathematical calculation to obtain satisfying results. Deep learning

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

strategies in machine learning, such as convolutional neural networks, recurrent neural networks, and recurrent neural networks, can produce excellent results due to recent advancements in processing power. Character recognition frameworks that are effective for a range of scripts have been developed, and these frameworks are based on deep neural networks (DNNs).

2. Related Work

It is possible to attribute the ever-increasing significance of hand digit recognition in the contemporary world to the widespread implementation of the technology in our day-to-day activities [1]. The past several years have seen the development of various recognition frameworks across many applications to improve order efficiency. Consequently, issues of varying degrees of complexity manifest themselves [2]. Hand-written digit recognition [3] is used rather frequently at the post office, where it is used to prepare bank checks in an automated fashion. In this piece of research, we constructed models for recognizing the digits 0 through 9 when they are written by hand using both artificial neural networks and convolutional neural networks. A neuron in the brain can be compared to a node in a neural network since they both play similar roles in the nervous system. The algorithm generates a well-balanced network by linking each node to other nodes using weights (which represent the edges between the nodes). A value is assigned to each node in the network based on the properties and operations of the nodes that came before it in the network. This approach is referred to as forward propagation in [6]. The loss function is utilized to adjust the weights to determine whether or not the system has been effectively theorized [7]. This occurs when the final output of the system is linked to the objective output. This procedure is referred to as "back propagation" [8,], a technical term. Layers of diverse types are used in neural networks to ensure that the networks are both complicated and accurate. A neural network that is fully connected will have several layers in its middle, including levels that are hidden and layers that produce information. Consider a list of characteristics with the labels x1, x2, x3,....., xn. Both forward and reverse propagation strongly rely on the weights allocated to the edges of the network that connect the different nodes. Before passing on the feature and the weights to the neuron or node, the hidden layer in a network that uses forward propagation will perform two unique operations on both the feature and the weights. The product of the features and weights includes an activation function as an additional component. When working with a deep neural network, you will find many weights and bias parameters to adjust. During backward propagation, it is possible to reduce the lost data by modifying the weights of prior time periods. In a neural network that is completely connected, the nodes of each layer are connected to the nodes of the layer below as well as the layer above [9].

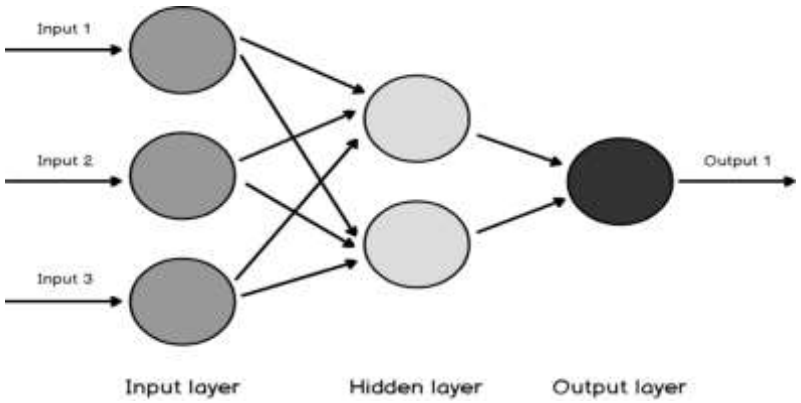

Figure 1: Artificial Neural Network

During the period spanning from 1980 to 2000, researchers encountered challenges in their attempts to successfully construct a functional deep neural network within an artificial neural network. The utilization of the sigmoid function is prevalent in each neuron, primarily due to the absence of the rectified linear unit (ReLU) until its inception in the year 2000. A vanishing gradient problem would be observed in a neural network. When applied to the summation of weights and features, the Activation function (specifically, the sigmoid function) maintains a consistent value ranging from 0 to 1. Additionally, the derivative of the activation function exhibits a value between 0 and 0.25, with a diminishing trend as the number of layers in the neural network increases. One potential way to address the issue of vanishing gradients is to employ the Rectified Linear Unit (ReLU) activation

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

function or another alternative activation function that does not lead to the diminishing of the derivative.

$$w_{new} = w_{old} - \eta * \delta_{loss}/\delta_{w_{old}} \quad (1)$$

When the weights assigned are large numbers, then the expected number of the derivative $\delta_{loss}/\delta_{w_{old}}$ will be a very large number, which will result in a significant variation in the new and old values when backpropagating. Then, new weight will jump on large values over the epochs, and the weights will vary a lot, with the value never converging at a point. So, the weight initialization in a Neural network is crucial; otherwise, this can lead to an Exploding Gradient problem [10].

There is a direct correlation between the amount of weights and bias factors and the depth of the neural network used. Overfitting the dataset or a subset of the data may become a concern. Underfitting is highly improbable in a multilayer Neural Network since many layers try to optimize their model to the training data. As the number of tiers in a network grows, the problem of excessive variation becomes increasingly pressing. Overfitting can be reduced with the help of regularization strategies like L1 or L2 regularization or the addition of Dropout layers. To use the Random Forest technique, many decision trees must be constructed. Overfitting is a potential problem while building decision trees because they are limited in depth. Like the decision tree, we'll use a particular class of features called regularization to improve the model's precision.

Selecting a small number of features from the input layer and a small number of hidden neurons is standard procedure when working with Neural Networks. Neurons not included in the selection process are silenced [11]. The dropout ratio is used to calculate the subset count of nodes. Image classification, object recognition, and data augmentation are just a few uses for Convolutional Neural Networks (CNNs). The input data for a Convolutional neural network (CNN) is typically a matrix with values between 0 and 255 in each cell. The convolutional neural network (CNN) uses a different number of artificial neural networks (NNs) depending on whether the input data is in grayscale (one NN) or RGB (three NNs) [12]. The procedure described here involves applying filters to photos and producing a matrix. Convolution layers with filters, pooling, a fully connected layer, and the application of a softmax function are only a few of the steps in a pipeline that the images travel through. A convolutional neural network is constructed to process incoming images and classify them according to specified values, as shown in the accompanying diagram [13].

3. The OCR system

Optical character recognition (OCR) is a complex recognition procedure that involves multiple steps. Every phase plays a critical role in shaping the overall framework of the model. Figure 2 illustrates the sequential arrangement of the processes in the Recognition model's pipeline.

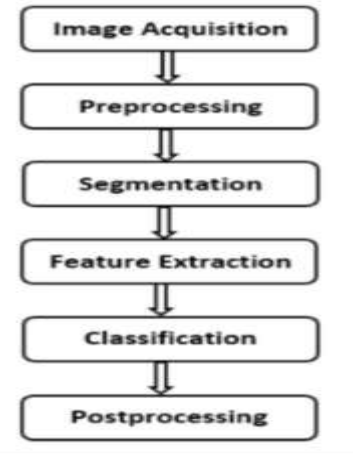

Figure 2: Character Recognition Stages

A. Dataset

The MNIST is a useful dataset to use when attempting to solve the challenge of classifying handwritten digits, as seen in Figure 3. The MNIST dataset is an extremely useful resource for students

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

and scholars. In total, 60,000 photographs have been sorted into 10 distinct groups (0–9). Every one of the images in the MNIST collection is made up of 784-dimensional vectors and has a height and weight of 28, respectively. The MNIST dataset is readily available for speedy download via the internet. A scale ranging from 0 to 255 indicates the brightness and darkness of each pixel that makes up an MNIST image. The MNIST dataset was generated by the NIST, which stands for the National Institute of Standards and Technology. To evaluate the model's potential, first the training set must be partitioned into a testing dataset. Plotting the outcomes of the model's execution on both the training and testing datasets is one way to get a visual representation of how far along the model is in its quest to solve the problem.

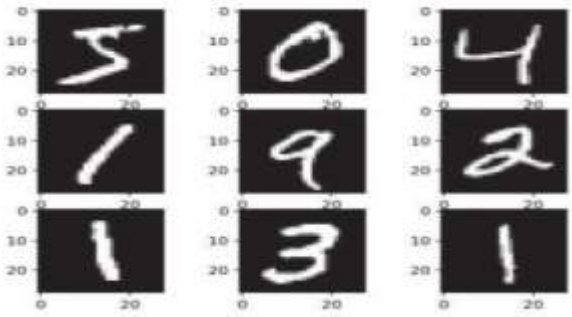

Figure 3: MNIST dataset

B. Image Acquisition

All recognition models begin with an image collection phase. In this step, the images are collected, filtered, and cleaned before being subjected to preprocessing.

C. Preprocessing

The "image preprocessing" stage is a very important part of the total process. Garbage collection and image cleaning to reduce noise levels are the two key responsibilities of the pre-processing stage. At this stage, the image is optimized by getting rid of any blank spots and smoothing out any curves that could be present. In addition, skew correction is performed, and several different methods are used. Binarization and texture filtering are used in this context, and the combined effect produces a binary image as the result.

D. Segmentation

When a single image is cut up into multiple smaller images, this process is known as segmenting the image. The three primary types of segmentation are known as linear segmentation, lexical segmentation, and morphological segmentation. The process of line segmentation begins by taking as input an image that contains many lines, which is subsequently segmented into a single line. Word segmentation is utilized when there is only a single line in the input image but many words must be separated. Words are segmented into their parts in a manner analogous to characters in character segmentation.

E. Feature Extraction

The "Feature Extraction" phase of the recognition model is critical. This is a crucial part of several methods for decreasing the dimensions involved. By reducing the dimensions, raw data can be simplified for further analysis and use. Large datasets, such as MNIST, are appropriate for usage in this stage because they are responsible for optimizing the recognition process as a whole. At this point, we have successfully deleted redundant data while maintaining the original dataset's integrity. Edge detection is just one of many image processing processes that improve upon the feature extraction phase. If feature extraction is neglected, classifying images becomes marginally more challenging and time-consuming. Techniques such as principal component analysis and image pixel vector can be used for feature extraction.

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

F. Classification

During picture recognition, the classification step is when the ultimate decision is made. As its source of information, this step uses the outcomes of the feature extraction step. Many different classifiers, such as logistic regression, random forests, k-nearest neighbors, support vector machines, convolutional neural networks, artificial neural networks, and support vector machine methods, can be used nowadays. Some of these classifiers are available for usage. Deep neural network classifiers, such as the Artificial Neural Network or the Convolutional Neural Network, can produce very satisfying results when used to categorize photos. Classifiers such as an Artificial Neural Network (Artificial neural network) or a Convolutional Neural Network (CNN) produce outstanding results when trained on 80% of the MNIST dataset, 10% of the dataset is used for validation, and 10% of the dataset is used for testing. This distribution of the dataset is known as the "80-10-10 rule."

4. Classifier

In this particular instance, we will be talking about Artificial Neural Networks. This model was conceived of and constructed inside of a computer. That is dependant on the structure as well as the components of the typical neurological framework. No matter how it takes place, data transmission will always result in structural changes to an artificial neural network. Consequently, the adjustments made to the neural framework depended on the information and the results. It attempts to replicate the pattern recognition capabilities of the brain. The principal use of neural networks is in resolving difficulties associated with layout. The feed-forward method became widely used once the multilayer perceptron model of artificial neural networks became prominent. The artificial neural network model comprises a network, and the weights of the edges connect the various nodes, also known as neurons. The weights allow us to decide which input class to use. The network's fourth-stage recognition model is the source of the information or features used by these stages. An artificial neural network can have an inappropriately low number of layers or an excessively big number of layers. Overfitting happens when there are excessive layers and neurons, and this issue needs to be resolved by either normalization or dropout. Jeffrey Hinton established the incredible approach of backpropagation and is a wonderful tool that opens up new doors for deep learning. As a result of this, we can construct a neural network that contains a deep layer. The gradient descent method is utilized in the backpropagation process to ensure that the weights of the edges and bais are kept up to date.

When it comes to the algorithms that are used in deep learning, convolutional neural networks are among the most successful models. When it comes to the challenge of categorizing images in computer vision, convolutional neural networks excel. Feature Extraction and Classification processes serve as the Primary Building Blocks for Convolutional Neural Networks. Deep learning is a type of machine learning in which algorithms are designed in a manner that is similar to how the human brain works. To learn a particular thing, just as it is necessary to train a Convolutional Neural Network with the help of photographs, so too do we need a collection of those images. Because numbers are the only language convolutional neural networks understand, an image must be encoded as a two-dimensional matrix (containing its pixels) before it can be trained. The convolutional neural network does not have an architecture comparable to an artificial neural network. In a Convolutional Neural Network, the model's layers have an architecture that makes them wide, tall, and deep. This is because each layer is composed of several smaller layers. The convolution layer of a convolutional neural network is analogous to the artificial neural network layer in that there is no communication between any of the layers in this layer. After processing the data, a convolutional neural network will produce an output in the form of a vector, the dimensions of which will be the probability scores. Following the pooling layer and the last fully connected layer is the convolutional layer, the final layer. A filter and a kernel work together in the first layer of a convolutional neural network to produce a feature map, which is then used for feature extraction when it has been completed. In the Convolutional layer, the convolution operation is carried out by first performing multiplication in each cell and then adding the results together on the feature map. The output can be nonlinear thanks to our activation function, which is a ReLU or Sigmoid. The number of steps the convolution filter takes at each iteration is referred to as the stride of the filter. Padding is utilized to prevent the feature map from further contracting due to its size, which is often smaller than the input. In most cases, the output of a convolutional neural network is a vector that indicates the requirement for dimensionality reduction. This is because the convolutional neural network often receives very large input. This issue is managed by inserting a pooling layer in the middle of a convolutional neural network. Utilizing the pooling layers in convolutional neural network models allows for effective overfilling management in these

Doi: https://doi.org/10.54216/JAIM.060102

Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

models. The two possibilities are maximum pooling and minimal pooling. The classification stage of a convolutional neural network model is the responsibility of the fully-connected layer. The convolution layer and the pooling layer transfer their output to the fully connected layer, so classification can occur. In the vast majority of instances, a fully connected system will only be able to process data in a single dimension. The artificial neural network and the convolutional neural network have a very similar final layer that does the convolutional processing. The topmost layer is fully connected, indicating that all of its components' connections are fairly robust.

5. Results

The digits 0 through 9 make up the classes in the classification process that make up the digit recognition section of the MNIST dataset. The Python programming language has an integrated development environment (IDE) called PyCharm, which is cross-platform and highly customizable. PyCharm is used with the latest stable version of Python, 3.7. The five steps of the recognition model are as previously mentioned: The implementation of data gathering has already occurred because the MNIST is a reliable dataset. To reduce the dataset's complexity, it is necessary to make all of the photos identical during the Image Processing phase of the Artificial Neural Network. The data is loaded by the Numpy library, which is included in the standard Python release and used extensively in scientific computing. As previously discussed, there are three levels to the Model of Artificial Neural Networks. There are multiple input and output layers in addition to a hidden layer. The information from one layer is fed into the next as input. Each layer of a Neural Network receives an input image whose size is proportional to the number of neurons in that layer. In our description, we referred to the dataset as 28x28, which is 784 pixels. In our model, the ReLU activation function is used to calculate the output of each layer. The output of each layer is determined by calling this function. The input and hidden layers maintain the same number of neurons throughout the training process. Since there are ten classes in the MNIST dataset (from 0 to 9), the output layer contains ten neurons for each class. Table 1 shows the outcomes of employing the CNN model shown in Figure 4, which was constructed with 15310 parameters. Accuracy during training and validation are shown in Figure 5, and training and validation losses are shown in Figure 6. The CNN model's confusion matrix is depicted in Figure 7.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_8 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_8 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 conv2d_9 (Conv2D)           (None, 6, 6, 40)          11560

 max_pooling2d_9 (MaxPooling  (None, 2, 2, 40)         0
 2D)

 flatten_5 (Flatten)         (None, 160)               0

 dense_10 (Dense)            (None, 20)                3220

 dropout_2 (Dropout)         (None, 20)                0

 dense_11 (Dense)            (None, 10)                210

=================================================================
Total params: 15,310
Trainable params: 15,310
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: CNN model

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

Table 1: Performance of CNN model

	accuracy	loss
0	0.82221669	0.530932903
1	0.925516665	0.226891667
2	0.942250013	0.179121777
3	0.949616671	0.155432984
4	0.955116689	0.139205873
5	0.959833324	0.126291454
6	0.96238333	0.119919017
7	0.966849983	0.106951602
8	0.97026664	0.095021307
9	0.97328335	0.086589895

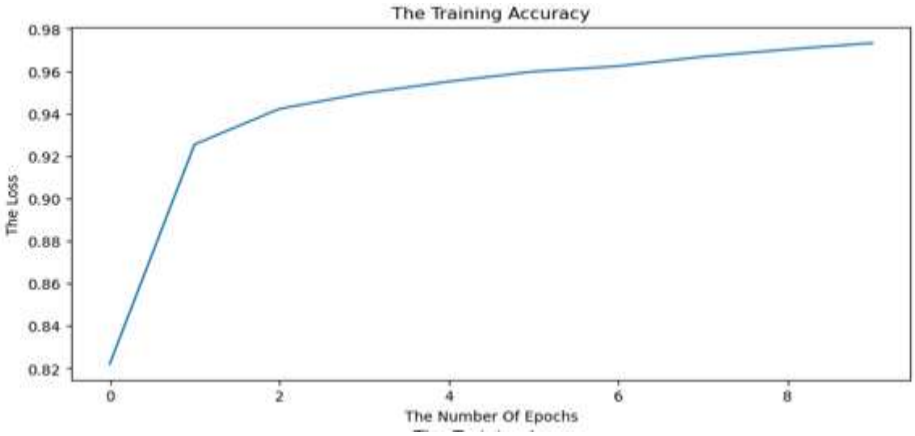

Figure 5: The training accuracy and the validation accuracy

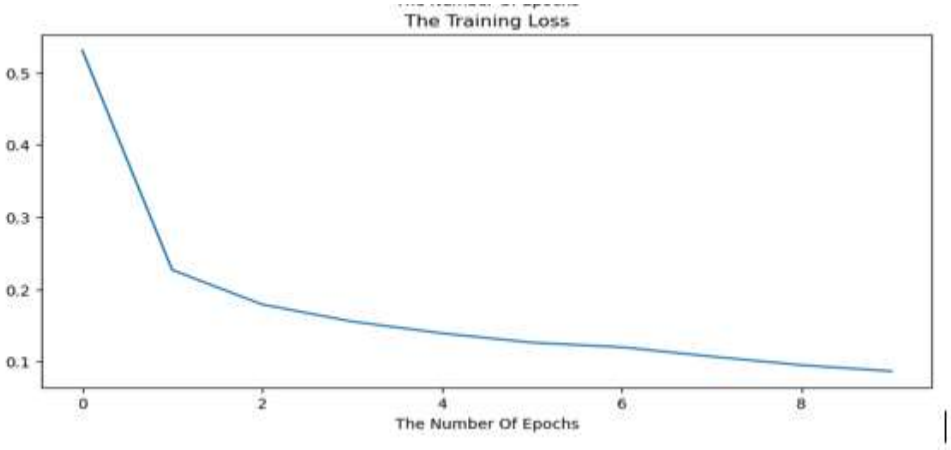

Figure 6: The training loss and the validation loss

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

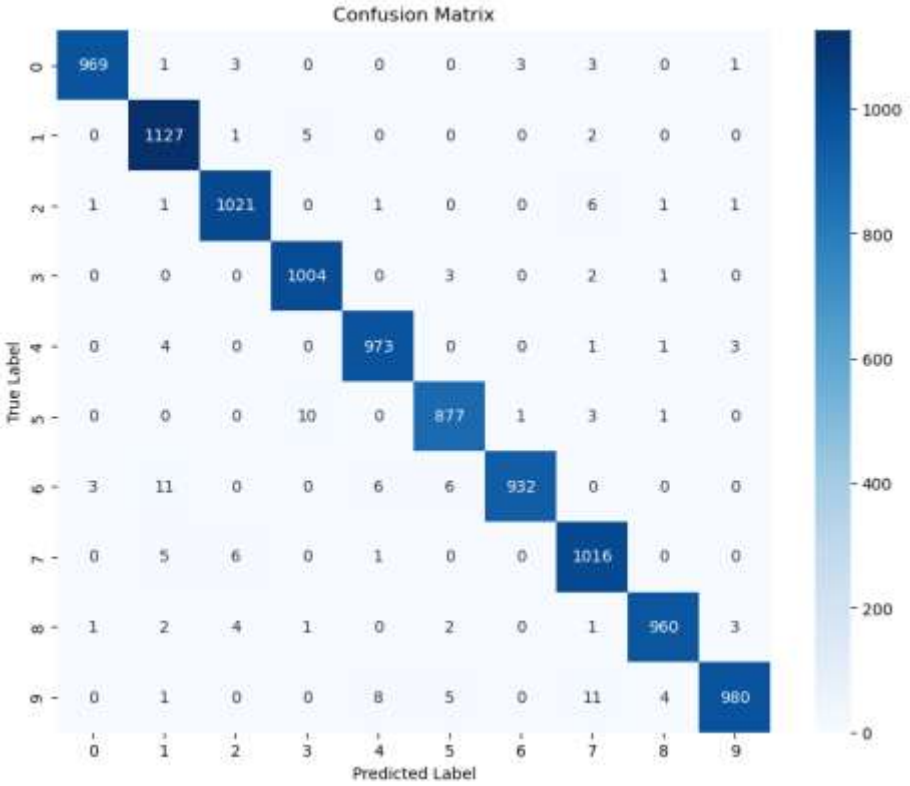

Figure 7: Confusion matrix of CNN model

6. Conclusion

Both convolutional neural networks and artificial neural networks can be trained and tested on the data provided by the MNIST dataset. Both models were trained on eighty percent of the data, and the remaining twenty percent was used to evaluate them. The ReLU activation function was utilized to assist in the training of the backpropagation method utilized by the models. In addition, a softmax method is used to activate the probabilistic values displayed as the output. For the sake of training in this instance, the categorical cross-entropy loss was applied. To improve the process even more, the ADAM optimizer was used to reduce the loss. During training for hardware reasons, it became apparent that CPU-based artificial neural networks had a higher error rate than convolutional neural networks, which are a type of neural network based on deep learning. The utilization of convolutional neural networks resulted in an improvement in the accuracy of the image categorization process. Artificial networks, on average, had a baseline error of 1.31 percent, but Convolutional Neural Networks were only wrong by 0.91 percent. This demonstrates why convolutional neural networks are preferable to artificial neural networks in this application. The processing time and hardware resources required by convolutional neural networks are significantly higher than those required by artificial neural networks. Regarding the rate at which computations may be completed, GPUs are superior to CPUs. Because more robust and cutting-edge processors are ideally suited for applying convolutional neural networks. The technique described above has a good chance of proving to be the most helpful resource for anyone interpreting handwritten digits. We can conclude from these findings that the technique taken by the convolutional neural network is superior to other methods in use today.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023

References

[1] M Nagu, N V Shankar, K Artificial neural network apurna, A novel method for Handwritten Digit Recognition with Neural Networks, 2011.

[2] Y LeCun, B E Boser, J S Denker, D Henderson, R E Howard, W E Hubbard, et al., Handwritten digit recognition with a backpropagation network, in Advances in neural information processing systems, 396-404, 1990.

[3] A Ashworth, Q Vuong, B Rossion, M Tarr, Q Vuong, M Tarr, et al., Object Recognition in Man, Monkey, and Machine, Visual Cognition, 5, 365-366, 2017 .

[4] J Janai, F Güney, A Behl, A Geiger, Computer Vision for Autonomous Vehicles: Problems, Datasets, and State-of-the-Art. arXiv preprint arXiv:1704.05519, 2017.

[5] K .Islam and R .Raj, Real-Time (Vision-Based) Road Sign Recognition Using an Artificial Neural Network, Sensors, 17, 853, 2017 .

[6] D Arpit, Y Zhou, B Kota, V Govindaraju, Normalization propagation: A parametric technique for removing internal covariate shift in deep networks, in International Conference on Machine Learning, 2016.

[7] I Patel, V Jagtap, O Kale, A Survey on Feature Extraction Methods for Handwritten Digits Recognition, International Journal of Computer Applications, 107, 2014.

[8] I H Witten, E Frank, M A Hall, C J Pal, Data Mining: Practical machine learning tools and techniques: Morgan Kauf Artificial neural network, 2016.

[9] Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Neural Network Algorithm Md.Abu Bakr Siddique; Mohammad Mahmudur Rahman Khan; Rezoana Bente Arif; Zahidun Ashrafi,2018

[10] Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with VariousHidden Layers and Epochs using Convolutional Neural Network Rezoana Bente Arif; Md Abu Bakr Siddique; Mohammad.Mahmudur.Rahman… Khan., Mahjabin Rahman. O, 2019

[11] Beyond human. Recognition: A.Convolutional neural network-based framework for .handwritten character recognition Li Chen; Song.Wang; Wei Fan; Jun Sun; Satoshi Naoi,2020

[12] Dhanya Sudarsan; Shelbi Joseph, A Novel Approach for Handwriting Recognition in Malayalam Manuscripts using Contour Detection and Convolutional Neural Nets, 2020.

[13] Nanehkaran, Y.A., Zhang, D., Salimi, S. et al. Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits. J Supercomput (2020).

[14] S. Oktaviani, C. A. Sari, E. Hari Rachmawanto and D. R. Ignatius Moses Setiadi, Optical Character Recognition for Hangul Character using Artificial Neural Network, 2020 International Seminar on Application for Technology of Information and Communication (semantic), Semarang, Indonesia, 2020.

[15] R. Sharma, B. Kaushik, and N. Gandhi, Character Recognition using Machine Learning and Deep Learning - A Survey, 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2020.

[16] P. Gupta, S. Deshmukh, S. Pandey, K. Tonge, V. Urkunde and S. Kide, Convolutional Neural Network-based Handwritten Devanagari Character Recognition, 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, 2020.

[17] P. Dhande and R. Kharat, Recognition of cursive English handwritten characters 2017 International Conference on Trends in Electronics and Informatics (ICEI) pp. 199-203 2017.

[18] Shalini Puri and Satya Prakash Singh, An efficient Devanagari character classification in printed and handwritten documents using SVM, Procedia Computer Science, 152, 111-121, 2019.

[19] J. Schmidhuber, Deep learning in neural networks: an overview. Neural Networks, 61, 85-117 2015.

Doi: https://doi.org/10.54216/JAIM.060102
Received: February 09, 2023 Revised: May 17, 2023 Accepted: November 15, 2023