International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Handwritten Digit Recognition of MNIST dataset using Deep Learning Convolutional Neural Network (CNN)

Franklin Robert Viegas
Department of Computer Engineering
PVPIT
ronyveg@gmail.com

Anjali Arvind Bodke
Department of Computer Engineering
PVPIT
anjalibodke2022@gmail.com

Paridhi Sudhir Wasnik
Department of Computer Engineering
PVPIT
paridhiwasnik23@gmail.com

Avinash Manojan Choran
Department of Computer Engineering
PVPIT
choranavinash@gmail.com

Under the Guidance of
Aeman Patel
aemanpatel86@gmail.com

***

Abstract - With applications ranging from postal services to digitized document processing, handwritten digit identification is a key problem in the fields of machine learning and computer vision. By utilizing their capacity to automatically extract hierarchical characteristics from unprocessed pixel data, Convolutional Neural Networks (CNNs) have become effective instruments for addressing this job. This study presents a thorough investigation of a CNN-based method that uses the MNIST dataset to recognize handwritten numbers. We explore the design, implementation, and performance assessment of the CNN model, demonstrating its ability to achieve high accuracy on tasks involving the recognition of numbers. We also go over the significance of our results for the larger picture of image categorization and suggest directions for further investigation and advancement.

Key Words: CNN, MNIST, Convolutional Layer, Pooling layer, Max Pooling, Neural Networks, Preprocessing, Dropout Layer, Activation layer, Rectified Linear Unit (ReLU), Epochs, MNIST dataset

INTRODUCTION

In the fields of machine learning and computer vision, handwritten digit recognition is a classic topic that attracts a lot of interest from both researchers and practitioners. Precisely recognizing and categorizing handwritten numbers is extremely useful in a variety of contexts, from automatic mail sorting to optical character recognition (OCR) in digital documents. This problem has been tackled in several ways over time, from more complex deep learning methods to more conventional machine learning algorithms. The ability of Convolutional Neural Networks (CNNs) to automatically train discriminative features directly from raw pixel input has made them the dominant paradigm among them. In this research, we provide an extensive analysis of a CNN-based method for handwritten digit recognition using the popular MNIST dataset as a reference. We shed information on the CNN model's design, specifics of its implementation, and performance assessment, emphasizing its effectiveness in reaching cutting-edge accuracy on the digit identification test.

.

RECENT WORK

1. "Handwritten Digit Recognition Based on Convolutional Neural Network", Kun He

Computers can be trained to understand handwritten numbers in pictures, even if they're written in different ways. This is called handwritten digit recognition, and it's an important part of computer vision. This ability is useful in many areas, like banking, online services, schools, mailing systems, and even transportation. This research paper describes a system for recognizing handwritten digits that uses a special kind of neural network called a convolutional neural network (CNN). The system is trained on a large dataset of handwritten digits called MNIST. In tests, the system was able to recognize the digits correctly almost 98% of the time, which shows that it works very well.

2. "MNIST Handwritten Digit Recognition using Machine Learning" Elizabeth Rani. G; Sakthimohan. M; Abhigna Reddy. G; Selvalakshmi. D; Thomalika Keerthi;Raja_Sekar.R

This work tackles the challenge of handwritten digit recognition (HDR) by utilizing the MNIST dataset. MNIST is a collection of labeled grayscale images, typically 28x28 pixels in size, where each image depicts a handwritten digit ranging from 0 to 9. Our approach leverages a Convolutional Neural Network (CNN) as the machine learning model for image classification. The core objective lies in training this CNN to achieve high accuracy in classifying the handwritten digits within the MNIST dataset. This research has the potential to significantly improve efficiency and reduce errors in automated tasks requiring HDR, such as automated check processing in the banking sector.

3. "Handwritten Digit Recognition using Convolution Neural Networks", Shailesh S Rajput; Yoonsuk Choi

Recent advancements in high-performance computing and neural network research have fueled a surge in deep learning capabilities. This paper explores core deep learning concepts and challenges encountered during model training. It proposes solutions for improved accuracy, exemplified by a convolutional neural network for digit recognition and prediction. The research delves into existing model training techniques for neural networks and sheds light on prior studies in digit recognition. Subsequently, it details a novel model for enhanced digit recognition accuracy. This model leverages sample preprocessing and ensemble learning, combining two or more models with diverse architectures and preprocessing methods. The implementation utilizes Python for both sample preprocessing and the construction of the model architecture.

4. "Digit Recognition by the Implementation of Supervised Learning Using a Convolutional Neural Network", Mrinal Paliwal; Sunil Kumar Chawla; Punit Soni Keerthi; Raja Sekar. R

Handwritten digit recognition remains a hurdle in computer vision, impacting applications in banking, medical diagnosis, and handwriting recognition. Technological advancements, fueled by improved internet access and developments in the field, have significantly bolstered our capabilities. Deep learning algorithms have become powerful tools for tackling various challenges, including digit recognition. Convolutional neural networks (CNNs) are particularly adept at this task due to their ability to extract hierarchical features from image data. This paper delves into the application of CNNs for digit recognition within machine learning. We explore the architecture of CNNs, their training processes, and various digit recognition methodologies. Additionally, we examine the datasets commonly used for training and testing CNNs in this domain, along with a discussion of results from prior research.

5. "Recognition of Handwritten Digit Using Neural Networks", Kancherla Santoshi; Chunduru Anilkumar; Suvvari Sravanthirani Bhardwaj

The identification of manuscript numbers is gaining importance due to its real-world applications. Several techniques, like calligraphy number recognition, are employed in various fields demanding high classification accuracy. Pattern recognition, a core component of computer vision and artificial intelligence, has been extensively used by researchers to develop practical applications such as automated bank check number reading. This project explores the use of Artificial Neural Networks (ANNs) with a focus on developing a system capable of recognizing handwritten numbers using two hidden layers. The backpropagation algorithm is employed to optimize the network weights, minimizing the overall error. ReLU activation functions are utilized in the hidden layers, while the output layer employs SoftMax activation. To train, validate, and test the model, we leveraged the publicly available MNIST handwritten digit dataset. A total of 60,000 digit images were obtained from the MNIST database for this purpose.

6. "Evaluating the Performance of Deep Learning Models in Handwritten Digit Recognition", Madan Lal Saini; Bharat Tripathi; Mohammed Sohail Mirza

Handwritten digit recognition plays a crucial role in computer vision, with applications in areas like bank check processing, postal service automation, and historical document digitization. Deep learning has emerged as a powerful tool for tackling this challenge. This study investigates four prominent deep learning architectures: LeNet-5, AlexNet, VGG-16, and ResNet-101, for handwritten digit recognition. We compare and contrast their performance using the MNIST, DIDA, and MNIST MIX datasets, aiming to identify the model offering the best balance between accuracy and efficiency. The paper provides a concise overview of each architecture, highlighting its unique design aspects and potential benefits. To ensure consistency and improve model generalization, the datasets undergo preprocessing. Evaluation metrics include accuracy, precision, recall, F1-score, and a visual inspection of model predictions. While all four architectures achieve significantly higher accuracy than traditional methods, their performance in terms of computational complexity, training time, and generalization capabilities varies. LeNet and ResNet demonstrate strong accuracy, while AlexNet and VGG exhibit faster training times.

7. "Recognition of Scanned Handwritten digits using Deep Learning", Paras Nath Singh; Kiran Babu T S

Handwritten digit recognition (HDR) is a challenging task for machine learning models due to the inherent variability in human handwriting styles. These variations can include inconsistencies in size, shape, and presence of noise or blur. To address these hurdles, this paper proposes a Convolutional Neural Network (CNN)-based deep learning approach for HDR. The proposed method preprocesses input images by converting them to grayscale and normalizing pixel intensities between 0 and 255. The CNN is then trained to classify these preprocessed images and recognize the digits they contain. To evaluate the effectiveness of our approach, we tested the model on the MNIST dataset, achieving an accuracy of 99%. Our method demonstrates the ability to recognize handwritten digits with diverse writing styles and variations in size. The model implementation utilized TensorFlow with Keras, a popular deep-learning framework in Python. Scaling the training data resulted in a slight decrease in accuracy, reaching 98.7%.

8. "Evaluation of Supervised Machine Learning Models for Handwritten Digit Recognition", Rohit Chandra Joshi; Vivek Raj Patel; Anjali Goyal

Handwritten document processing is a critical task due to the sheer volume of documents and associated processing costs. Handwritten digits, a frequent component in documents, pose a particular challenge as individual writing styles can lead to recognition difficulties. This paper explores the use of machine learning for handwritten digit recognition (HDR). Applications of HDR include automatic license plate recognition, postal sorting systems, and check processing. To address the inherent variability in handwriting, we conduct an empirical analysis comparing various supervised machine-learning algorithms for HDR. These algorithms include Naive

Bayes, k-nearest Neighbors, Logistic Regression, Support Vector Machines, Random Forests, Gradient Boosting, Convolutional Neural Networks, and Decision Trees. We evaluate their performance using metrics such as accuracy and F1-score to identify the most effective algorithms for HDR tasks.

9. "KNN and the CNN for Handwritten Digit Recognition: A comparative study", Taher Mostafa El-Sahhar; Mohamed A. Wahby Shalaby

Handwritten digit recognition (HDR) technology allows computers to automatically identify handwritten digits. It holds promise for applications like processing bank documents, financial statements, and postal mail. This study investigates the use of K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN) for HDR using the MNIST handwritten digit database. To optimize performance, we implemented KNN in Python and CNN in TensorFlow during the training phase. The research compares the advantages and disadvantages of these two AI-powered techniques for handwritten digit recognition, evaluating their performance through recognition rate.

10. "Handwritten Character and Digit Recognition with Deep Convolutional Neural Networks: A Comparative Study", Chui En Mook; Chin Poo Lee; Kian Ming Lim; Jit Yan Lim

In handwritten character or digit recognition (HCCR), the goal is to automatically categorize handwritten characters or digits from images. Prior research often concentrated on individual datasets and lacked in-depth comparisons between various CNN architectures. This paper bridges this gap by presenting a comparative analysis of six prominent CNN architectures (VGG16, Xception, ResNet152V2, InceptionResNetV2, MobileNetV2, and DenseNet201) applied to three distinct datasets: English handwritten characters, handwritten digits, and MNIST. The experimental results reveal that the InceptionResNetV2 model, coupled with data augmentation techniques, achieves superior accuracy across all datasets. It attained accuracies of 93.26%, 97.16%, and 99.71% on English handwritten characters, handwritten digits, and MNIST datasets, respectively.

PROPOSED CNN MODEL.

The layers of CNN receive the input image. The required features are extracted from the input image by these CNN layers through training. Each layer carries out three processes: convolution, activation, and the creation of smaller images. The first convolutional layer uses a 32, (3×3) size filter kernel, whereas the second convolutional layer uses a 64, (3×3) size filter kernel. Next, a non-linear Rectified Linear Unit (ReLU) is used to perform the activation operation. Ultimately, a pooling layer that recognizes the various sections of the image creates a reduced-size copy of the original image.

Furthermore, the model has a dropout layer with a 0.25 dropout rate that works to prevent overfitting by randomly setting a portion of the input units to zero during training. We are employing a stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 to develop the complete model, using accuracy as the parameter to track throughout training. Categorical cross-entropy is the loss function, appropriate for multi-class classification. With 421,642 trainable parameters in all, our CNN model is the last one. A dataset of 28x28 grayscale photos with 10 distinct classes may now be used to train this model.

Let's dissect each component of the model:

1. Input Layer: Using the ReLU activation function, the input layer ({Conv2D}) has 32 filters of size (3,3).
- The input form is (28,28,1), which represents single-channel, 28x28 pixel images in grayscale.

2. Pooling Layer: To downsample the spatial dimensions, use the MaxPooling layer with a pool size of (2,2).

3. The Second Convolutional Layer: This is an additional `Conv2D} layer featuring 64 filters of varying sizes (3,3), ReLU activation, and max pooling.

4. Dropout Layer: During training, a random portion of the input units are set to zero by the Dropout Layer, which has a dropout rate of 0.25 and helps to prevent overfitting.

5. Flatten Layer: - This layer flattens the convolutional layers' 2D output into a 1D vector that is fed into the dense, fully linked layers.

6.DenseLayers: 128 units in the first dense layer with ReLU activation. A dropout layer has a rate of 0.5 dropouts.

7. Output Layer: - Final Dense layer, appropriate for multi-class classification issues, with softmax activation and 10 units (matching to the number of classes in the output).

8. Model Compilation: - This step involves putting the model together using a stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and categorical cross-entropy as the loss function (fit for multi-class classification). Accuracy is the statistic to watch throughout training.

9. Model Summary: - Every layer in the model is described in depth in this summary, including the type of layer, the output shape, and the quantity of trainable parameters. There are 421,642 trainable parameters in the model overall. Currently, a dataset of 28x28 grayscale photos with ten distinct classes can be used to train this model.

HARDWARE AND SOFTWARE REQUIREMENTS

Operating System: Windows 11

Memory: 16 GB RAM

CPU: 8 core, 3.40 GHz processor.

GPU: Nvidia RTX 2060

Software: Python is a basic programming language that is useful

for Artificial Intelligence applications due to its stability and versatility. Python's NumPy library is used for scientific computing and data analysis, making data manipulation and storage more efficient than with the language's built-in data structures.
Computer vision applications are facilitated by OpenCV, an open-source, cross-platform library. To provide high-level interfaces for capturing, processing, and presenting the image data, it is written in optimized C++.
A consistent format for data that is compatible with scientific libraries like NumPy is provided by OpenCV's ability to bind with Python.
TensorFlow, an open-source library from Google, is used for sophisticated numerical calculations and large-scale machine learning.

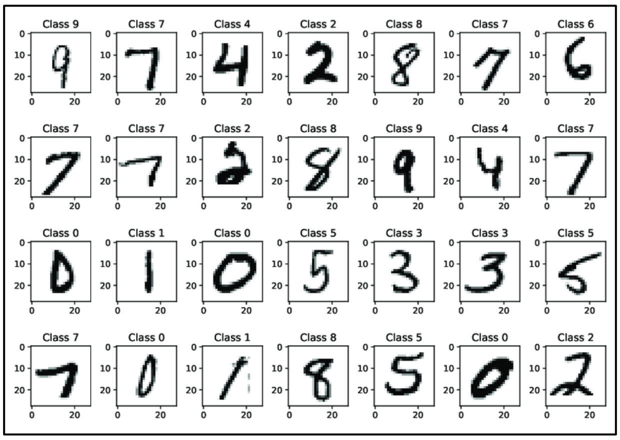

Fig.-1 Dataset

MNIST DATASET

Many image processing algorithms are trained using data from the MNIST (Modified National Institute of Standards and Technology) database, which is a sizable collection of handwritten numerals or digits. The dataset is also frequently used in the field of machine learning for testing and training. Two of NIST's databases, Special Database 1 and Special Database 3, are combined to create the collection of photographs in the MNIST database.

There are 10,000 test images and 60,000 training images in the MNIST dataset.

The MNIST dataset, which can be accessed online, is essentially a collection of different handwritten numbers. A popular dataset for showcasing the true potential of deep neural networks is the MNIST dataset, which contains a lot of data. Any numbered image can be recognized by our eyes and brain working together. Our minds are powerful tools that can classify any image in a short amount of time. A number can take on a variety of shapes, and while our minds are adept at identifying these shapes and identifying the number they represent, computers struggle to accomplish the same task. This can only be accomplished by training a computer to accurately classify handwritten digits through the use of deep neural networks.

TensorFlow's MNIST dataset splits handwritten digit data into three sections:
o Training Data: 55000 datapoints (mnist.train)
o Validation Data: 5000 datapoints (mnist.validate)
o Test Data: 10000 datapoints (mnist.test)

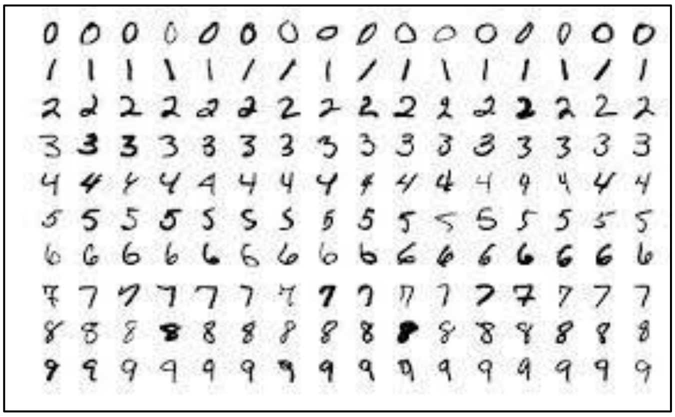
Fig-2 MNIST Dataset

WORKING

The proposed system has multiple modules, namely Pre-processing, Feature Extraction, MinMaxScaler -fitting of data, Image normalization, and Classification.

A. PRE-PROCESSING: To do pre-processing on an input image, first convert it to grayscale. The three channels that make up a typical colored image are red, green, and blue, or RGB for short. After that, the colored image is transformed into a grayscale image with a single monochrome channel to remove any extraneous noise. When the input image is compared to that of a trained convolutional neural network, the difference in the size of the given image could result in a loss of accurate prediction. Therefore, to make the image resolution equal to that of the EMNIST dataset, it is scaled and placed on top of a blank, 28 by 28-pixel image.

B. FEATURE EXTRACTION: The process of turning a set of features that accurately describes the input data from the input is known as feature extraction. Dimensionality reduction and feature extraction are connected. It is possible to decrease the amount of input data into a smaller set of features (also known as a feature vector) when it is too big to process. Feature selection refers to choosing a subset of the original features. To accomplish the intended goal utilizing this reduced representation rather than the whole starting data, the chosen features are anticipated to contain the pertinent information from the input data. Pixel values are obtained in the form of a 1D array, which indicates values between 255 and 0 based on pixel intensity after the image has been resized.

C. MIN MAX SCALER: All of the data are boxed into a range between a given min and max value using the mean and standard deviation in the min-max scalar method of normalization. Each feature is scaled to a specified range to alter the features. To get each feature on the training set to fall inside the specified range—that is, between zero and one—this estimator separately scales and translates each feature. Scaling with unit variance and zero mean is frequently substituted with this transformation. It makes the range smaller such that it now lies between 0 and 1 (or, in the case of negative numbers, -1 and 1). Arguably the most well-known scaling algorithm is the MinMaxScaler, which uses the following formula for each feature: (min(x)–xi) / (max(x)–min(x))

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

D. IMAGE NORMALIZATION: The process of normalization modifies the pixel intensity value range. Histogram stretching or contrast stretching are other names for normalization. The character will be delivered exactly as it appears in the image once the background pixels in this input image are removed during normalization. To achieve this, use a random value to ensure that the background pixels' value is significantly lower than the character's shade pixel values. The image is normalized in this way to make it resemble the values in the EMNIST dataset. Where the character "A" is written in this image, the pixel values are greater than 0, while all other sections have pixel values of zero after normalization.

E. CLASSIFICATION: The handwritten character is classified from the input image using a convolutional neural network as the classifier. A CNN is made up of several hidden layers in addition to its input and output layers. Generally speaking, convolutional, pooling, fully connected, and normalizing layers make up a CNN's hidden layers. The pooling layer, output layer, and convolutional layer are the three main parts of a CNN. The activation function known as ReLU, or Rectified Linear Unit, is frequently utilized with CNN. Each neuron in the convolution layer that is connected to a local region in the input volume will compute its output by taking the dot product of its weights and that local region.
Nonlinear downsampling is what the pooling layer does. With max pooling, the most popular method, the input image is divided into a collection of non-overlapping rectangles, and the maximum is output for each of these sub-regions. ReLU utilizes the activation function that is not saturated. Without changing the convolution layer's receptive fields, it enhances the nonlinear features of the decision function and the network as a whole. When the input value is less than zero, the rectified linear unit produces 0; otherwise, it produces raw output. The following formula is used to determine its value: $f(x)= \max(x,0)$ A neural network-based classifier's last layer frequently uses the softmax function.
The softmax function works similarly to a sigmoid function in that it squashes the outputs of each unit to be between 0 and 1. Additionally, each output is divided so that the total of all the outputs adds up to 1. A categorical probability distribution can be represented by the softmax function's output. Therefore, across a set of 'n' events, the softmax function computes the probability distribution of each event.

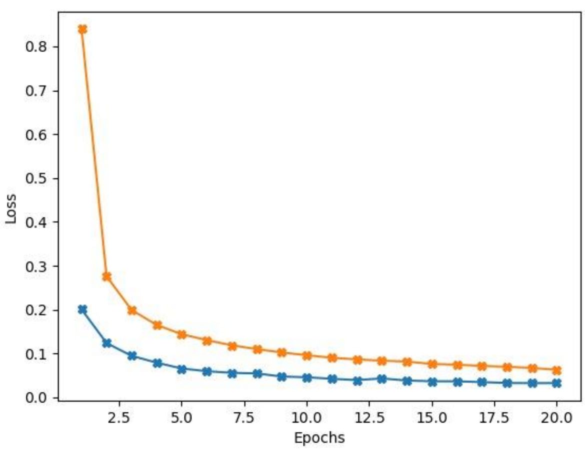

Fig-3 Loss vs Epochs Graph

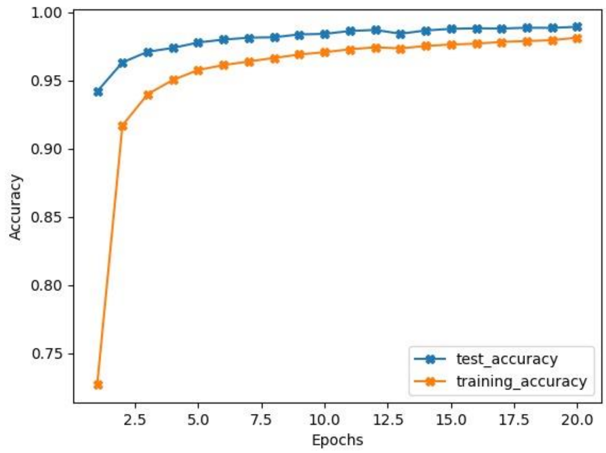

Fig-4 Accuracy vs Epochs Graph

RESULT AND ANALYSIS

This section includes experimental results and performance assessments of our suggested methodology using handwritten samples from real-world scenarios as well as established benchmark datasets. We perform extensive experiments to evaluate our handwritten character recognition system's scalability, resilience, and accuracy.

Evaluation matrix:
To evaluate the effectiveness of our handwritten character recognition system, we use common evaluation measures, such as the following:

1. Accuracy: The proportion of characters in the dataset that are properly recognized over all characters in the dataset.

2. Precision, Recall, and F1-Score: Based on the confusion matrix of predicted and ground truth labels, these metrics are used to assess how well multi-class classification tasks are performed.

3. Word Error Rate (WER): The ratio of the total number of insertions, deletions, and substitutions needed to match the predicted and ground truth text sequences is used to measure the accuracy of text recognition algorithms.

Our findings show that, when applied to a variety of datasets and handwriting styles, our suggested methodology delivers great accuracy and robustness.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
EMNST	98.94	98.94	98.94	98.94

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

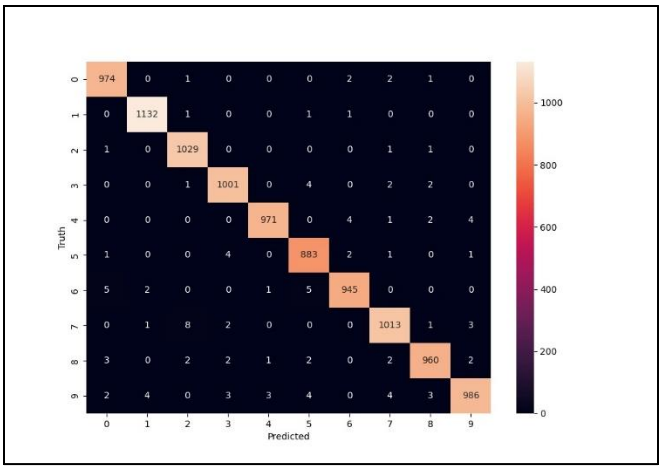

Fig-5 Confusion matrix

CONCLUSION

In summary, this paper has provided an in-depth analysis of a Convolutional Neural Network (CNN) method for handwritten digit recognition with the MNIST dataset. By testing and analyzing large amounts of test and training data, we have shown that our CNN model performs far better than other models when it comes to correctly categorizing handwritten numbers. Our results highlight how well deep learning methods—in particular, CNNs—work when applied to actual picture categorization issues. Our research establishes the groundwork for future studies and innovations in the field of handwritten digit recognition, opening doors for developments in OCR systems, pattern recognition, document processing, and other related fields. We can open up new avenues for using technology to solve challenging real-world problems and raise the effectiveness and precision of automated systems by constantly pushing the limits of machine learning and computer vision.

REFERENCES

1. K. He, "Handwritten Digit Recognition Based on Convolutional Neural Network," 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2023, pp. 16-19. doi: 10.1109/ICETCI57876.2023.10176680.

2. E. R. G, S. M, A. R. G, S. D, T. Keerthi and R. S. R, "MNIST Handwritten Digit Recognition using Machine Learning," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 768-772. doi: 10.1109/ICACITE53722.2022.9823806.

3. S. S. Rajput and Y. Choi, "Handwritten Digit Recognition using Convolution Neural Networks," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2022, pp. 0163-0168. doi: 10.1109/CCWC54503.2022.9720854.

4. M. Paliwal, S. Kumar Chawla and P. Soni, "Digit Recognition by the Implementation of Supervised Learning Using a Convolutional Neural Network," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023, pp. 227-231. doi: 10.1109/ICTACS59847.2023.10389920.

5. K. Santoshi, C. Anilkumar and S. Sravanthirani Bhardwaj, "Recognition of Handwritten Digit Using Neural Networks," 2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur, Odisha, India, 2022, pp. 1-5. doi: 10.1109/iSSSC56467.2022.10051289.

6. M. Lal Saini, B. Tripathi and M. S. Mirza, "Evaluating the Performance of Deep Learning Models in Handwritten Digit Recognition," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023, pp. 116-121. doi: 10.1109/ICTACS59847.2023.10390027.

7. P. N. Singh and K. Babu T S, "Recognition of Scanned Handwritten digits using Deep Learning," 2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon), HASSAN, India, 2023, pp. 1-6. doi: 10.1109/MysuruCon59703.2023.10396920.

8. R. C. Joshi, V. R. Patel and A. Goyal, "Evaluation of Supervised Machine Learning Models for Handwritten Digit Recognition," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 378-383. doi: 10.1109/ICAC3N56670.2022.10074292.

9. T. M. El-Sahhar and M. A. W. Shalaby, "KNN and the CNN for Handwritten Digit Recognition: A comparative study," 2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2023, pp. 339-342. doi: 10.1109/NILES59815.2023.10296578.

10. C. E. Mook, C. Poo Lee, K. M. Lim and J. Yan Lim, "Handwritten Character and Digit Recognition with Deep Convolutional Neural Networks: A Comparative Study," 2023 11th International Conference on Information and Communication Technology (ICoICT), Melaka, Malaysia, 2023, pp. 137-141. doi: 10.1109/ICoICT58202.2023.1026272