



MACHINE LEARNING APPROACH FOR HANDWRITTEN CHARACTER RECOGNITION AND CLASSIFICATION

Dr.G.NallaSivan¹, professor, CSE Department, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu
A.Nithin², Student (Final year), CSE Department, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu
T.Surya³, Student (Final year), CSE Department, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu
D.Hanuram Surya, Student (Final year), CSE Department, PSN College of Engineering and Technology, Tirunelveli, Tamilnadu

ABSTRACT

The machine learning-based handwritten digit identification system suggested in this research study is a model that can be used to detect and recognize digits written by a user on an editable canvas widget inside a Graphical User Interface (GUI). This research article describes handwritten digit recognition systems in depth, our strategy to creating one, and analyzes the accuracy of several machine learning methods that may be used to create such systems.

Keywords: Handwritten Digit Recognition, GUI, Machine Learning Algorithm, Handwritten Digit Recognition System, Accuracy, Convolutional Neural Network, Support Vector Machine.

I. INTRODUCTION

Because it is a highly practical technique, handwritten digit identification is one of the most essential topics that should be tackled in the area of machine learning. There are several handwritten digit recognition applications, such as check processing in banks, data input in forms, mail sorting, and so on.

Machine learning is one of the most important ideas to highlight in order to fulfill the ever-increasing everyday demands of the IT sector. We tackled the challenge of constructing a handwritten digit recognition system utilizing machine learning to better comprehend it and apply it to problem solving. This issue will enable us to master machine learning methods from the ground up, and it also includes a variety of application cases that may benefit people or businesses.

Machine learning encompasses several kinds of learning models. These are the following:

A. Supervised Education

Naive Bayes, K-Nearest Neighbors, Random Forest, Decision Trees, Linear Regression, and Support Vector Machine are some of the algorithms used in supervised learning.

During the training phase of supervised learning, a model is trained using a labeled dataset in which two variables, called 'input' and 'output,' are mapped to each other.

B. Unsupervised Learning

Unsupervised learning methods include Neural Network, Anomaly Detection, K-Mean Clustering, and Multivariate Analysis.

During the training phase in unsupervised learning, the model is trained using an unlabeled dataset, whether classified or unclassified. During the learning phase, input is not transferred to output; instead, the training input dataset is classified into classes, which are then used to predict the output for the testing dataset.

C. Reinforcement Learning

Reinforcement learning methods include Negative, Positive, Q Learning, and Markov Decision Process.

Decisions are made progressively in reinforcement learning. An intelligent agent behaves on the basis of collective benefits in order to maximize these rewards.

II. MNIST DATASET

Yann LeCun, Corinna Cortes, and Christopher Burges created the MNIST dataset, which stands for Modified National Institute of Standards and Technology. Handwritten digit recognition systems utilizing machine learning methods have long been popular among academics.

The scanned digits were standardized in size and justified as centered, making it an ideal database for testing these models. The error rate may be greatly lowered by utilizing separate classifiers for different methods and parameters. As training examples, the MNIST collection contains a variety of handwritten digits. It comprises of 70,000 photos, 60,000 of which are used in the training dataset and 10,000 of which are used in the testing dataset, both of which include suitably labeled images of the digits 0-9. Handwritten digits are displayed as grayscale pictures of 28*28 pixels. Every MNIST data point consists of two components: a target label associated with the handwritten digit and a picture of the handwritten digit itself. When utilizing the MNIST dataset, very little data cleaning is necessary, allowing one to concentrate



completely on the goal of his/her machine learning/deep learning model..

Fig 1: Various styles of handwritten labels within the dataset

III. LITERATURE SURVEY

Machine learning methods may be used to develop handwritten digit recognition systems. The time necessary to train the model and the final accuracy of the model that is assessed might vary when various techniques are employed to tackle this particular challenge. In general, there is a time-accuracy tradeoff when developing such systems using various techniques. Some techniques take less time to train the model entirely than others, but they yield poorer accuracy as a consequence. On the other, several alternative techniques require a lengthy time to thoroughly train the model yet eventually result in more accurate results.

Because this technology has many applications, some of them demand better precision to operate well and have no time constraints, whilst others may require the model to be trained quicker but with a bigger margin of error.

Because the outcomes of various algorithms differ in terms of time and accuracy, they may be compared appropriately. Our study will focus on accuracy, and we will compare several algorithms on the same. To construct a handwritten digit recognition system, we recommended using the advantages of a Convolutional Neural Network. We trained our model with numerous convolution and pooling layers to achieve high accuracy. Our suggested work necessitates the precise identification of digit strings in order to transform the recognized decimal number to the binary/octal/hexadecimal number system. According to our findings, CNN is the best potential algorithm for reaching maximum accuracy.

A. Existing System

- Proximal Support Vector Machine (PSVM), Multilayer Perceptron, Support Vector Machine (SVM), Random Forest, Bayes Net, Naive Bayes, J48, and Random Tree are some of the techniques used to create handwritten digit recognition systems.
- According to past research, these algorithms yield accuracies on the order of: Proximal SVM - 98 percent Multilayer Perceptron - 90 percent
- 87 percent for SVM
- Random Forest had an 85 percent success rate, Bayes Net had an 84 percent success rate, and Naive Bayes had an 81 percent success rate.

Even while these algorithms may be effective in certain applications based on this technology, many other applications, such as banking sector applications, need superior outcomes that may be delivered utilizing alternative algorithms than the ones listed above.

B. Proposed System

Convolutional Neural Network (CNN) may be used to create handwritten digit recognition systems in order to minimize error and increase overall efficiency. To do this, our suggested system employs CNN with numerous pooling and convolutional layers, as well as a 3x3 kernel. During the training phase, our model employs 60,000 28*28 grayscale photos. Our model is trained over a normal 5 epochs to reach accuracy of the order of 99.16 percent, which is much greater than the usual techniques used to create handwritten digit recognition systems such as SVM, Multilayer Perceptron, Bayes Net, Random Forest, and so on.

IV. PROPOSED APPROACH

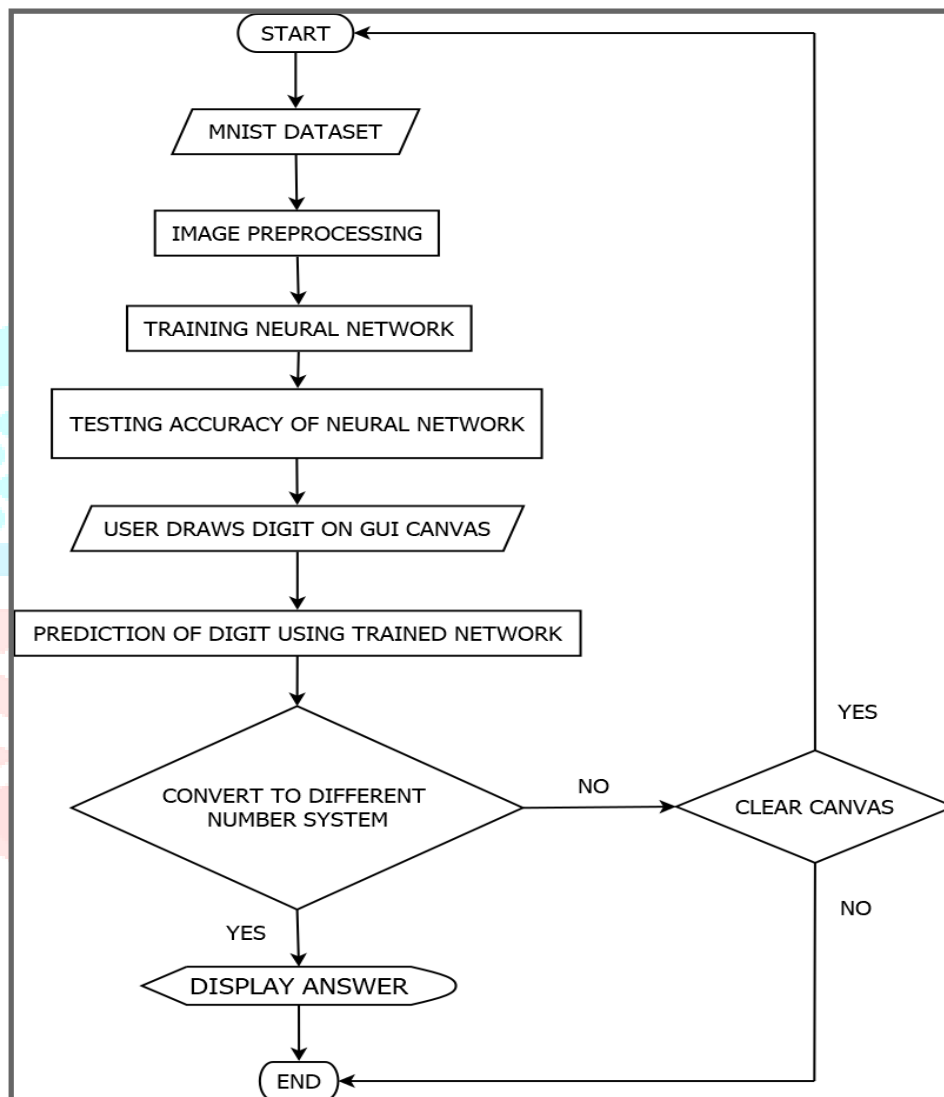


Fig 2: Activity diagram of our proposed project

The suggested work aims to detect user-defined handwritten digits and identify the whole number (by default in decimal number system) that is input by the user, which is then converted to binary/octal/hexadecimal number system based on the user's preference. For the same, a graphical user interface (GUI) will be constructed in which the user will be presented with a canvas widget that he or she may use to design handwritten digit strings for recognition and conversion. The canvas may then be cleared for future development.

A. Dataset

The suggested model is trained using the MNIST dataset. It is made up of 70,000 digital photos that may be used to train and evaluate the model. These training and testing datasets are designed around a specified ratio. This picture data is then cleaned and preprocessed in preparation for future processing.

B. Image Preprocessing

Image preprocessing is the process of using numerous approaches such as scaling pictures, converting them to

grayscale format, and image augmentation in order to properly employ digital image data inside the machine learning model.

C. Training Neural Network

Following the completion of data preprocessing, the CNN model will be generated, which will include several convolutional and pooling layers as well as a 3x3 sized kernel. The model will then be trained using training and validation data from multiple Python libraries, including TensorFlow, Pillow, OpenCV, Tkinter, and Numpy, which were preloaded to accomplish these particular tasks.

D. Testing Accuracy of Neural Network

We utilize the testing dataset to assess the model's performance after it has been trained using the training dataset. A subset of the whole MNIST dataset is utilized as the testing dataset, from which the suggested model's accuracy is calculated.

E. User Draws Digit on GUI Canvas

The trained model is then employed through a Graphical User Interface (GUI) based canvas where a user creates digits using the mouse pointer by clicking and dragging the mouse suitably once the model has been assessed, i.e. trained and tested using the MNIST dataset.

F. Recognize Number/Clear Canvas

- Following the user's selection of numbers on the GUI canvas, the user is presented with two options:
- Recognize Number: This option use the CNN model to anticipate the user's string of numbers.
- Clear Canvas: This option enables the user to clear the canvas and draw additional numbers to continue the drawing.

G. Convert to Different Number System

- Following the digit prediction, the user is presented with three options:
- Convert to Binary: The Convert to Binary option transforms the recognized decimal number to its binary counterpart.
- Convert to Hexadecimal: Converts a recognized decimal number to its hexadecimal counterpart.
- Convert to Octal: Converts the recognized decimal number to its octal counterpart.

V. IMPLEMENTATION DETAILS

A. Digit Recognizer File

The MNIST dataset will be used to develop the handwritten digit recognition system needed for this project. The MNIST dataset will be put into our digit recognition Python script to do this. Following that, a sequential CNN model will be built, to which convolution and pooling layers will be added. The digital picture data will be filtered using a 3x3 kernel. Following the pooling and convolution layers, the data is transformed using the 'flatten' function to turn the multidimensional data input into a single dimension in order to shift to a fully connected layer. The suggested CNN model will be trained using the 'relu' activation function.

Following that, the picture data is binarized, or changed from 28*28 grayscale image format to binary format matrix. Following that, the MNIST dataset is separated into 60,000 training samples and 10,000 testing samples. Finally, the model is built by training it over 5 epochs using the 'rmsprop' optimizer and a batch size of 64. The suggested CNN model's loss and accuracy are then tested, and the model is stored for further use in the GUI python file.

B. GUI File

The saved model is first imported into the GUI python file, and then the primary GUI window displaying the canvas widget is generated using the 'Tk' function. For the master window, a mainloop is established, which runs indefinitely until the user closes the window. The primary GUI window is then given a title linked to the proposed project. The primary window has two buttons: 'Recognize Number' and 'Clear Canvas.' Following that, the functions for implementing functionality such as cleaning the canvas, drawing digits, activating the event for doing so, and digit recognition are defined.

To recognize the handwritten user-defined digit strings on the canvas displayed within the GUI, a list of contours is created, which is very useful for detecting the digit and analyzing its shape. A contour is a line that connects every point around the borderline of an image that has similar intensity.

The 'Recognize Number' button is then hit, causing the model to anticipate each digit one by one. It presents the result in a new window, where each number is identified independently, as well as the precision with which they are detected. This new window is given the needed title and has three additional possibilities in addition to the recognized number.

These three choices, in the end, give the feature of converting the recognized decimal number to the binary/hexadecimal/octal number system of the user's choosing.

VI. LIMITATIONS

One of the most challenging aspects of developing a handwritten digit identification system is that there are several diverse handwriting styles, which is a very personal activity. Numbers may have distinct stress portions, be written at different angles, and have varied lengths of certain segments. Although machine learning developers face these challenges, several steps have already been taken, such as fine tuning already defined models and developing cutting-edge classification methods for predicting handwritten digits effectively by reducing computational cost, time, and also improving accuracy. Extensive study is also being undertaken in this sector to ensure its long-term viability.

If this concept is applied on a large scale, many complications may occur. Recognizing handwritten numerals, if used maliciously, might lead to a number of problems. People may use such technology to identify bank pins, ATM pins, and so forth in order to commit monetary crimes. On the contrary, even if such issues arise, steps can be taken to effectively address them and sustain the use of this technology to automate many processes such as banking, address recognition, shipping systems, the postal industry, and so on, making it beneficial to use because it will ultimately have more pros than constant.

VII. CONCLUSION

Based on our findings, we may conclude that Machine Learning algorithms function very well at detecting trends in various writing styles. Handwritten digits may be recognized using a variety of techniques. CNN delivers the highest level of accuracy for recognizing/predicting handwritten digits. By deleting the ensemble features and fine-tuning the hyper parameters of the pure CNN architecture, the accuracy of these classic CNNs may be increased even more. This reduces the computational difficulties as well as the total cost of implementing the model.

This technique may also be used on GPUs alongside CPUs to boost efficiency by lowering calculation time. Compute Unified Device Architecture (CUDA) on a GPU shortens training time, increasing the overall efficacy of the proposed machine learning model. When it comes to the future potential of this machine learning-based technology, it is incredibly adaptable and has several applications in the banking, postal, and shipping sectors, where it may allow the automation of many procedures. Some of them include identifying the amount and account details on cash slips to automate and save time in the banking process. Address recognition software may also be used to help the postal and shipping industries. Overall, this technology may be utilized to serve a wide range of complicated applications, some of which may or may not be real-time.

VIII. REFERENCES

- [1] Swapna Prava Ekka, "Recognition of Handwritten Digits using Proximal Support Vector Machine", (2014), National Institute of Technology Rourkela, unpublished.
- [2] Viragkumar N. Jagtap, Shailendra K. Mishra, "Fast Efficient Artificial Neural Network for Handwritten Digit Recognition", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), (2014), ISSN Number: 0975-9646, pp. 2302-2306.
- [3] Saeed AL-Mansoori, "Intelligent Handwritten Digit Recognition using Artificial Neural Network", Journal of Engineering Research and Applications, Vol. 5, Issue 5, (Part -3) May (2015), ISSN : 2248- 9622, pp.46-51.
- [4] S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana, Abdullah Al Jobair, "Handwritten Digit Recognition using Machine Learning Algorithms", Journal of Computer Science and Technology: DNeural & Artificial Intelligence, vol. 18, Issue 1, (2018), ISSN: 0975-4172, pp.0975-4350.
- [5] Md. Anwar Hossain, Md. Mohon Ali, "Recognition of Handwritten Digit using Convolutional Neural Network (CNN)", Journal of Computer Science and Technology: D Neural & Artificial Intelligence, Vol. 19, Issue 2, (2019), ISSN: 0975-4172, pp.0975-4350.
- [6] Saqib Ali, Zeeshan Shaukat, Muhammad Azeem, Zareen Sakhawat, Tariq Mahmood, Khalilur Rehman, "An efficient and improved scheme for hand written digit recognition based on convolutional neural network", (2019), unpublished.
- [7] Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh, Byungun Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)", (2020), unpublished.
- [8] Aarti Gupta, Rohit Miri, Hiral Raja, "Recognition of Automated Hand-written Digits on Document Images Making Use of Machine Learning Techniques", Journal of Engineering and Technology Research, (2021), ISSN: 2736-576X.
- [9] V. Gopalakrishnan, R. Arun, L. Sasikumar, K. Abhirami, "Handwritten Digit Recognition for Banking System", Kings College of Engineering, Punalkulam, Pudukottai, Journal of Engineering Research & Technology (IJERT), (2021), ISSN: 2278-0181.
- [10] Ritik Dixit, Rishika Kushwah, Samay Pashine, "Handwritten Digit Recognition using Machine and Deep Learning Algorithms", Journal of Computer Applications, Vol. 176 – No. 42, July (2020), pp.0975 – 8887.
- [11] Kiran Banjare, Sampada Massey, "Handwritten Numeric Digit Classification and Recognition: Recent Advancements", Journal of Emerging Technologies in Engineering Research (IJETER), Vol. 4, Issue 6, June (2016), ISSN: 2454-6410.
- [12] Pranit Patil, Bhupinder Kaur, "Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models", Journal of Innovative Research in Computer Science & Technology (IJRCST), Vol. 8, Issue- 4, July (2020), ISSN: 2347-5552.
- [13] Bushra Aurangzeb, Memuna Malik, "A Survey on Handwritten Digit Recognition", COMSATS Institute of Information technology, Vol. 4, Issue 2, April (2015), ISSN: 2306-708X.
- [14] Narender Kumar, Himanshu Beniwal, "Survey on Handwritten Digit Recognition using Machine Learning", H.N.B. Garhwal University, Vol-6, Special Issue-5, June (2018), ISSN: 2347-2693.
- [15] Akanksha Gupta, Ravindra Pratap Narwaria, Madhav Singh, "Review on Deep Learning Handwritten Digit Recognition using Convolutional Neural Network", Journal of Recent Technology and Engineering (IJRTE), Vol. 9 Issue-5, January (2021), ISSN: 2277-3878.