Handwritten Digit Recognition System Using Machine Learning

Ujjwal Jindal*,Vikas Panwar†,Ujjwal Gupta‡, Gaurav Parashar§
Department of Computer Science and Engineering
KIET Group of Institutions
Ghaziabad, India
Email: *ujjwal.1923cs1059@kiet.edu,
†vikas.1923cs1013@kiet.edu, ‡ujjwal.1923cs1179@kiet.edu, §gauravparashar24@gmail.com (ORCID: 0000-0003-4869-1819)

***Abstract*—This research study proposes a model for a handwritten digit identification system based on machine learning which could be used to recognize and identify digits written by a user on a canvas (editable) widget inside a graphical user interface (GUI). The certainty of several machine learning algorithms which can be used to develop such systems is compared in this research study in conjunction with a complete discussion of handwritten digit recognition systems, the approach used by us to implement them, and their applications.**

***Index Terms*—handwritten digit recognition, GUI, machine learning algorithm, handwritten digit recognition system, accuracy, convolutional neural network, support vector machine.**

I. INTRODUCTION

As a very practical technique, handwritten digit recognition is one of the most crucial problems in machine learning that should be solved. It also plays an important role in knowing about the various fields that use pattern recognition. Applications for handwritten digit recognition include sorting mail, processing checks in banks, entering data onto forms, and more. Machine learning is one of the most essential ideas to be highlighted in order to meet the constantly expanding everyday demands related to the IT business. We set out to construct a handwritten digit recognition system using machine learning to grasp machine learning better and exercise it in problem-solving. This problem provides a number of use cases that could be advantageous for people or organizations, and it will allow us to study machine learning procedures from the basics. Machine learning includes a variety of learning model types. These are listed below: A. Supervised Learning Supervised learning consists of various algorithms some of which used are as Naive Bayes, Random Forest, K-Nearest Neighbors, Decision Trees, Support Vector Machine, and Linear Regression. In the supervised learning approach, a model is trained with the help of a labeled dataset which consists of two variables, referred to as "input" and "output," that are mapped to one another. B. Unsupervised Learning Unsupervised learning consists of Numerous techniques some of which are Neural Networks, K- Mean Clustering, Multivariate Analysis, and Anomaly Detection. When using an unsupervised learning strategy, the model is trained with the help of an unlabeled dataset, whether it is classified or not. During the learning process, input is not transferred to output; rather, the input dataset which is to be trained is sorted into groups, which helps to forecast the result of the testing dataset. C. Reinforcement Learning Different algorithms which come under reinforcement learning include Q Learning, Negative, Positive, and Markov Decision Process. Reinforcement learning uses Sequential decision-making. Based on collective incentives, an intelligent agent manipulates its surroundings in an effort to maximize these rewards.

II. MNIST DATASET

Yann LeCun, Corinna Cortes, and Christopher Burges created the modified National Institute of Standards and Technology dataset or MNIST dataset. Various digits which are scanned were standardized in size and justified as centered, which made it an exemplary database helping in the evaluation of these models. This made it particularly prominent among academics for building handwritten digit recognition systems utilizing machine learning techniques. The error rate can be greatly decreased by utilizing a variety of classifiers for distinct algorithms and parameters. The MNIST dataset consists of training examples that include a variety of digits that are handwritten. It includes 70,000 images, out of which 60,000 are used to train the model and 10,000 are used in the testing of the model. Both datasets contain snaps of the 10 digits, starting from 0 to 9, that have been accurately classified. The representation of handwritten numbers is a 28*28 grayscale picture. Each MNIST data point consists of two components: an image of the handwritten digit and a target label pertaining to it. When using the MNIST dataset, very little data cleaning is necessary, allowing one to concentrate completely on the goal of their deep learning model or machine learning model.

III. LITERATURE SURVEY

The handwritten digits are preprocessed in several standard databases, including segmentation and normalisation, so that researchers can compare the recognition outcomes of their techniques on an equal footing, according to Li Deng [1],

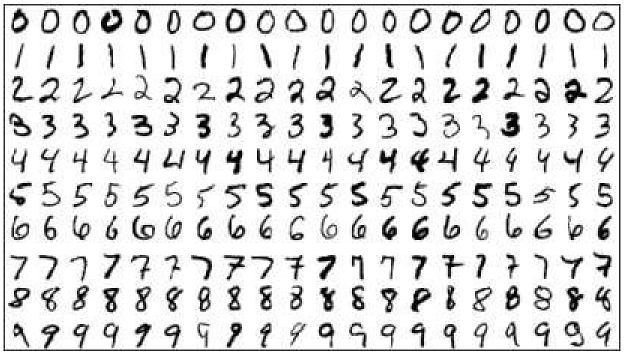

Fig. 1. Dataset shows the different designs of handwritten labels

who claimed that Handwritten Digit Recognition is one of the significant problems in the recognition of an optical character. He came to the conclusion that the MNIST database offers a relatively straightforward and static classification assignment for academics and students to investigate pattern recognition and machine learning since it eliminates the need for pointless data pretreatment and formatting.

By fitting generative models created using deformable B-splines with Gaussian ”ink generators” that are spaced along the length of the spline, M. Revow, C.K.I. Williams, and G.E. Hinton [2] investigated a technique for recognising handwritten digits. They later came to the conclusion that while this method requires high computation, it may be used as a verification stage for faster recognizers to achieve improved performance because there will be little correlation between Additionally, they have shown how generative models can be used to extract additional information from images that is useful for model-driven segmentation.

By making a number of changes to the previously described neural network model neocognitron, Kunihiko Fukushima [3] proposed an improved version and verified its effectiveness by employing a sizable database of handwritten digits (ETL1). The self-organization of line-extracting cells, the highest level of supervised competitive learning, the contrast-extracting layer followed by edge-extracting cells, the inhibitory surround in the connections between S- and C-cells, etc. are a few examples of the many alterations. Later, He came to the conclusion that neocognitron is a simpler network than other ANN. Additionally, the number of repeated presentations of a training set required by neocognitron is much lower than for the network trained by backpropagation, and thanks to the modifications proposed, the accessory circuits present in the earlier versions can be removed.

In order to improve the performance of the existing object and shape recognition model, Mandana Hamidi and Ali Borji [4] suggested adding more biologically inspired attributes as lateral inhibition, feature localisation, and feature sparsification. Later, they came to the conclusion that the modified model performs better than the original model at recognising English and Farsi handwritten digit datasets, and that it can also be used to identify individuals using their palm, iris, or fingerprint biometrics because these features have intricate structural details.

Using the spike-triggered Normalised Approximate Descent (NormAD) technique, Shruti R. Kulkarni and Bipin Rajendran [5] have demonstrated supervised learning in Spiking Neural Networks (SNNs) for the problem of handwritten digit identification. They have ended their effort by offering numerous experimental insights for the improvement of learning parameters and network setup. According to their experiments, even with 3-bit synaptic weights, the developed SNN's classification accuracy does not decrease by more than 1 percent in comparison to the floating-point baseline. With four times fewer parameters than the state-of-the-art network, their network—which uses neurons operating at sparse biological spike rates below 300 Hz—achieved a classification accuracy of 98.17 percent on the MNIST test database.

An adaptive deep Q-learning technique for handwritten digit recognition was put forth by Junfei Qiao [6]. The suggested technique, known as Q-ADBN (Q-learning Adaptive Deep Belief Network), combines the decision-making abilities of reinforcement learning with deep learning's capacity to extract features. The Q-ADBN initially utilises an adaptive deep auto-encoder (ADAE) to extract features from the input images before utilising the Q-learning method to determine the final recognition choice by maximising the Q-function. According to the experimental findings, the proposed Q-ADBN works better in terms of recognition accuracy and processing speed than existing approaches of a similar kind.

My Donell [7] discusses a study that demonstrated shallow non-convolutional neural networks trained using the ”Extreme Learning Machine” (ELM) technique may achieve error rates below 1 percent on the MNIST handwritten digit benchmark. On the NORB image database, it was demonstrated that the ELM approach could achieve less than 5.5 percent error rates. The conventional ELM algorithm has been improved by the authors in a number of ways that can greatly boost speed. The study also discovered that the ELM technique is suitable for many practical machine learning applications since it is simple to use and accurate when creating a single-hidden-layer neural network classifier. The study's findings indicate that the use of deep convolutional networks may involve confirmation bias, as opposed to straightforward single-layer feedforward.

An automated process for producing features for handwritten digit recognition is described by Gader, P.D.[8]. The technique directs the search for features using the two evaluation metrics orthogonality and information. High classification rates are achieved by using the generated features in a neural network that has been trained using backpropagation. On a test set of 1000 digits per class, the classifier is combined with several other high-performance classifiers to achieve recognition rates of about 98 percent.

A deep convolutional extreme learning machine (DC-ELM) technique for image classification problems is presented by Pang, Shan, and Xinyi Yang [9]. It combines the benefits of an extreme learning machine (ELM) and a convolutional neural network (CNN) to improve generalisation performance and speed up training. To extract high-level characteristics from raw input images, the approach uses a number of different convolution and pooling layers. These features are then given to an ELM classifier. In order to decrease feature dimensionality and conserve computational resources, the DC-ELM includes stochastic pooling. The DC-ELM performs better than existing ELM methods and cutting-edge deep learning approaches in tests on handwritten digit recognition tasks in terms of test accuracy and training time. Applications define the best network structure for DC-ELM, and there is a way to find the best structure.

He, Sheng, and Lambert Schomaker [10] offer a deep adaptive learning approach for single-word image-based writer identification. To enforce the emergence of reusable characteristics, the method entails including an auxiliary task during the training phase. To take use of the deep features that were learnt from the auxiliary task, the authors suggest a brand-new adaptive convolutional layer. Three different auxiliary tasks that correspond to the explicit information in handwritten word images are evaluated after the multi-task neural network has been trained from beginning to end. Results demonstrate that the suggested strategy outperforms non-adaptive and straightforward linear-adaptive alternatives in terms of writer identification performance.

A growing amount of interest has been generated in finding a solution to the open problem of handwritten digit recognition, which Ali Alani [11] have shown. Research on handwritten digit recognition in Arabic is scarce, despite the fact that many research have been proposed in the past and most recently to enhance handwritten digit recognition in a variety of languages. In the feature extraction step, we start by using the RBM, a deep learning technique that can extract extremely valuable features from raw data and has been applied to a number of classification issues. Last but not least, a comparison of our findings with those of other investigations on the CMATERDB 3.3.1 Arabic handwritten digit dataset demonstrates that our methodology the highest degree of accuracy.

Suiyang Khoo, Zhihong Man, Kevin Lee, Dianhui Wang, and Zhenwei Cao [12] This research develops an optimal weight learning machine for a handwritten digit image identification using a single hidden layer feedforward network (SLFN). It can be seen that the SLFN's input and output weights have both undergone global optimisation using the batch learning type of least squares. In order to maximise the separability of all nonlinearly separable patterns and achieve a high level of recognition accuracy with a minimal number of hidden nodes in the SLFN, all feature vectors of the classifier can then be positioned at the specified locations in the feature space. A test to see whether handwritten digit images can be recognized. The proposed methodology exhibits good performance and efficacy using both the MNIST database and the USPS database.

According to S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana, and Abdullah Al Jobair [13], one of the practically significant problems in pattern recognition applications is handwritten character recognition. Applications for digit recognition include filling out forms, processing bank checks, and sorting mail. The capacity to create an effective algorithm that can recognise handwritten digits and which is submitted by users via a scanner, tablet, and other digital devices is at the core of the issue. WEKA has been used to recognise digits using a variety of machine learning algorithms, including Multilayer Perceptron, Support Vector Machine, NaFDA5, Bayes, Bayes Net, Random Forest, J48, and Random Tree. The study's findings indicate that the top 90.37 percent accuracy has been obtained for Multilayer Perceptron.

This paper introduces a novel deep learning architecture called DIGITNET and a large-scale digit dataset called DIDA to detect and recognise handwritten digits in historical document images written in the nineteenth century, according to a proposal by Huseyin Kusetogullari, Amir Yavariabdi, Johan Hall, and Niklas Lavesson [14]. Digit ized from historical Swedish handwritten documents authored by many priests in a variety of handwriting styles are collected to create the DIDA collection. Three sub-datasets of this dataset are available: a single-digit subset, a large-scale bounding box annotated multi-digit subset, and a digit string subset containing samples in Red-Green-Blue (RGB) colour spaces. Additionally, DIDA is used to train the DIGITNET network, which consists of the DIGITNET-dect and DIGITNET-rec deep learning architectures, to isolate and identify digit strings in historical handwritten documents.

The adaptive function neural network (ADFUNN) and online snap-drift learning are combined in a novel way by Miao Kang and Dominic Palmer-Brown [15] and used for optical and pen-based recognition of handwritten digits. [E. Alpaydin, F. Alimoglu for Optical Recognition of Handwritten Digits and E. Alpaydin, C. Kaynak for Pen-Based Recognition of Handwritten Digits Snap-drift [S.W. Lee is a quick, unsupervised method appropriate for online learning and non-stationary situations where new patterns are continuously supplied. It combines the complimentary principles of common (intersection) feature learning (referred to as snap) and LVQ (drift towards the input patterns) learning.

A. Existing System

Some of the algorithms used to develop handwritten digit recognition systems are Random Tree, Support Vector Machine (SVM), Proximal Support Vector Machine (PSVM), Multilayer Perceptron, Random Forest, Bayes Net, Naive Bayes, and J48. Previous studies have shown that these algorithms provide accuracy around:
- Proximal SVM Algorithm - 97%
- Multilayer Perceptron Algorithm- 91%
- SVM Algorithm - 88%
- Random Forest Algorithm - 86%
- Bayes Net Algorithm - 85%
- Naive Bayes Algorithm - 82%
- J48 Algorithm - 80%
- Decision Tree Algorithm - 76%

Even while some applications based on this technology may find these algorithms valuable, many other areas of applications, like those in the Finance business, call for superior outcomes that could be obtained by utilizing alternative algorithms in comparison to the algorithms that have already been described.

B. Suggested System

Handwritten digit recognition systems can be developed by using Convolutional Neural Network (CNN) in order to lower error and increase overall efficiency. Our suggested approach makes use of a 3x3-sized kernel and CNN with numerous pooling and convolutional layers to accomplish this. In order to train our model, 60,000 28*28 grayscale photos are used. Our model is trained through a typical 5 epochs to attain correctness around 99.15% rather than conventional techniques used to develop handwritten digit recognition systems, like J48, Decision tree, Native Bayes, etc.

IV. SUGGESTED APPROACH

The aim of our proposed work is to identify handwritten digits defined by the user and recognize the full digit that the user inputs (by default, in the decimal number system), which is then translated to the user's preferred binary, octal, or hexadecimal number system. For this, a graphical user interface (GUI) will be developed in which a canvas widget will be visible to the user which will help him to draw handwritten digit strings that will be recognized and converted appropriately. After that, the area can be cleaned up to continue.

A. Dataset

The suggested model is trained using the MNIST dataset. 70,000 digital photos make up the collection, which may be used to train and evaluate the model. These datasets for training and testing are created using a particular ratio. Then, this image data is cleansed and prepared for advancement.

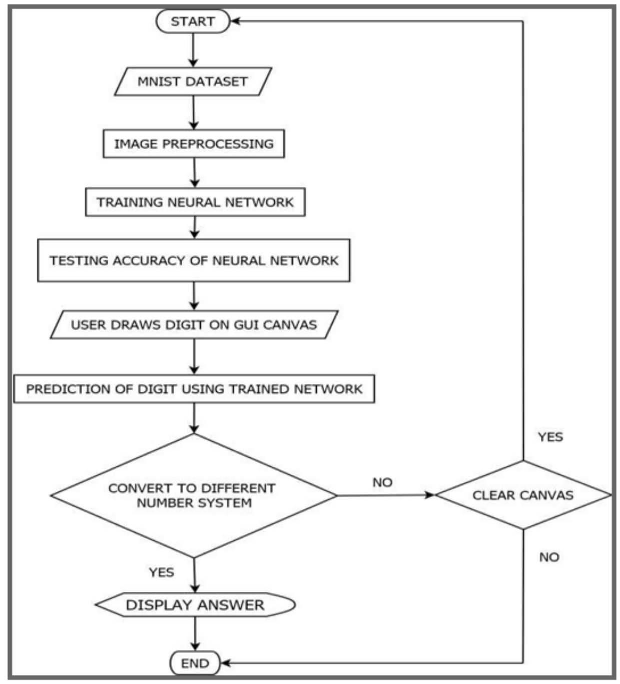

Fig. 2. Flowchart of Activities in Suggested project

B. Image Preprocessing

In this phase, several techniques are put into practice, like shrinking the pictures, transforming them into the format of grayscale, and enhancing the picture, in order to make the digital image data useful for our machine learning system.

C. Training Neural Network

The CNN model, which includes various pooling and convolutional layers along with a kernel of size 3x3, would be developed after data preprocessing is finished. The model will then be trained using training and validation data with the help of many pre-loaded libraries of Python, such as Scipy, Theano, TensorFlow, Keras, Pandas, Matplotlib, etc.

D. Testing Accuracy of Neural Network

Testing datasets are then used to gauge the model's effectiveness once it has been trained using the training dataset. MNIST's whole dataset is used to test the suggested model's accuracy.

E. User Draws Digit on GUI Canvas

Once the model has been reviewed, trained, and tested using MNIST dataset, it is then available for use via a Graphical User Interface (GUI) based canvas where a user can draw numbers using the mouse pointer.

F. Recognize Number/Clear Canvas

The user is presented with two alternatives after drawing the desired number of numbers on the GUI canvas-

- Recognize Numbers. In order to forecast the user's string of digits, this option makes use of the CNN model.

- Clear Canvas: With this option, the user may clear the canvas and add new digits to the existing ones.

G. Convert to Different Number System

The user is presented with three options following the occurrence of the predicted digits-

- Binary conversion: This option converts the recognized decimal number into binary.
- Hexadecimal conversion: This option converts the recognized decimal number to its hexadecimal.
- Octal conversion: This option converts the recognized decimal number to its octal.

V. IMPLEMENTATION DETAILS

A. Digit Recognize File

The handwritten digit recognition system needed for this project will be built using the MNIST dataset. To do this, the MNIST dataset will be added to our digit recognition Python application. The established sequential CNN model will next be extended with convolution and pooling layers. A 3x3-sized kernel will be used to filter the digital image data.
After the data has been processed through pooling and convolution layers, the 'flatten' function is used to reduce the multidimensional data input to a single dimension before moving to a fully connected layer. The activation function that will be utilised to train the suggested CNN model is called "relu".
Following this, the image data is binarized, or changed from its original 28*28 grayscale format to a matrix in binary. Then, 60,000 training samples and 10,000 testing samples are separated from the MNIST dataset. The final step in compiling the model is training it through 5 epochs using the 64-batch "rmsprop" optimizer.
The suggested CNN model's loss and accuracy are then assessed, and the model is saved for use in the GUI Python file at a later time.

B. GUI File

After the saved model has been loaded into the GUI Python file, the main GUI window that displays the canvas widget is initially created using the 'Tk' method. A main loop is given to the master window, and it keeps going until the user shuts it. The title of the main GUI window is then assigned to the proposed project. The "Recognise Number" and "Clear Canvas" buttons are situated in the main window.
The procedures for putting features into action, such as cleaning the canvas, drawing the numbers, initiating an event to do so, and identifying the numbers, are then defined. The border of an image is defined as a set of contours, or a line connecting all the points with a similar intensity.
The model begins to predict each and every digit one by one once you click the "Recognise Number" button. The result, which is given in a new window, recognises each digit separately and displays the accuracy with which it was detected. In addition to the three additional options and the indicated number, this new window also has the required title. The feature of translating the detected decimal number to the user's selected binary, hexadecimal, or octal number system is finally enabled by these three options.

VI. LIMITATIONS

The fact that there is so many various handwriting style which is a very personal behavior—makes developing a handwritten digit identification system one of the most challenging tasks. Numbers can be written at different angles, with varying amounts of stress, and with various segment lengths. Although machine learning developers encounter these difficulties, some steps have already been taken, such as fine-tuning already mentioned models and developing cutting-edge classification algorithms for accurately predicting handwritten numbers while decreasing computing cost and time. Additionally, extensive study is being done in this area to support it appropriately.

If this concept is used on a large scale, several problems might occur. If utilized maliciously, the ability to recognize handwritten numbers could eventually result in a number of problems. Such technology could be used by criminals to identify ATM, bank, and other types of pins. Contrarily, even though problems like these might arise, steps could be taken to address them and maintain the use of this technology to computerize a variety of tasks, including banking, address recognition, shipping systems, the postal service, and others, making it advantageous to work, as it will ultimately have more benefit than drawbacks.

VII. CONCLUSION AND FUTURE SCOPE

Our study has led us to the conclusion that machine learning algorithms are extremely effective at identifying trends throughout various writing styles. Handwritten digits can be recognized using a variety of techniques.
It may be said that CNN recognizes and predicts handwritten digits with the highest degree of accuracy. By removing the fine-tuning of the hyper parameters in the pure convolutional neural network (CNN) architecture, the accuracy of these conventional CNNs can be increased even further. This will help lower the total cost and complexity of the model's computations.
This technology can also be implemented on the GPU in addition to the CPU to increase efficiency by speeding up calculation. The suggested machine learning model performs better overall because training time is cut down by using Compute Unified Device Architecture (CUDA) on a GPU.
There are many applications of machine learning-based technology in the banking, shipping, and postal sectors, and it can automate many operations in these sectors. Some of these involve automating the banking procedure and saving time by examining cash slips for account numbers and amounts. The postal and shipping industries could potentially profit from computerized address recognition.

In general, this technology may handle a wide range of complicated applications, some of which are real-time in nature and some are not.

REFERENCES

[1] Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine, 29(6), 141-142.

[2] Revow, M., Williams, C. K., & Hinton, G. E. (1996). Using generative models for handwritten digit recognition. IEEE Transactions on pattern analysis and machine intelligence, 18(6), 592-606.

[3] Fukushima, K. (2003). Neocognitron for handwritten digit recognition. Neurocomputing, 51, 161-180.

[4] Hamidi, M., & Borji, A. (2010). Invariance analysis of modified C2 features: case study—handwritten digit recognition. Machine Vision and Applications, 21, 969-979.

[5] Kulkarni, S. R., & Rajendran, B. (2018). Spiking neural networks for handwritten digit recognition—Supervised learning and network optimization. Neural Networks, 103, 118-127.

[6] Qiao, J., Wang, G., Li, W., & Chen, M. (2018). An adaptive deep Q-learning strategy for handwritten digit recognition. Neural Networks, 107, 61-71.

[7] McDonnell, M. D., Tissera, M. D., Vladusich, T., van Schaik, A., & Tapson, J. (2015). Fast, simple and accurate handwritten digit classification by training shallow neural network classifiers with the 'extreme learning machine'algorithm. PloS one, 10(8), e0134254.

[8] Gader, P. D., & Khabou, M. A. (1996). Automatic feature generation for handwritten digit recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(12), 1256-1261.

[9] Pang, S., & Yang, X. (2016). Deep convolutional extreme learning machine and its application in handwritten digit classification. Computational intelligence and neuroscience, 2016.

[10] He, S., & Schomaker, L. (2019). Deep adaptive learning for writer identification based on single handwritten word images. Pattern Recognition, 88, 64-74.

[11] Alani, A. A. (2017). Arabic handwritten digit recognition based on restricted Boltzmann machine and convolutional neural networks. Information, 8(4), 142.

[12] Man, Z., Lee, K., Wang, D., Cao, Z., & Khoo, S. (2013). An optimal weight learning machine for handwritten digit image recognition. Signal Processing, 93(6), 1624-1638.

[13] Shamim, S. M. (2018). Handwritten digit recognition using machine learning algorithms. Global Journal Of Computer Science And Technology, 18(D1), 17-23.

[14] Kusetogullari, H., Yavariabdi, A., Hall, J., & Lavesson, N. (2021). DIGITNET: a deep handwritten digit detection and recognition method using a new historical handwritten digit dataset. Big Data Research, 23, 100182.

[15] Kang, M., & Palmer-Brown, D. (2008). A modal learning adaptive function neural network applied to handwritten digit recognition. Information Sciences, 178(20), 3802-3812.