HANDWRITTEN DIGIT RECOGNITION

K. Bavani
Associate Professor
Computer Engineering and Science
Krishnankoil, Tamil Nadu's Kalasalingam
Academy of Research and Education
k.bavani@klu.ac.in

M. J. Sai Kiran
Computer Engineering and Science
Krishnankoil, Tamil Nadu's Kalasalingam
Academy of Research and Education
9920004371@klu.ac.in

N. Vinay
Computer Engineering and Science
Krishnankoil, Tamil Nadu's Kalasalingam
Academy of Research and Education
9920004356@klu.ac.in

S. Tejeswar Reddy
Computer Engineering and Science
Krishnankoil, Tamil Nadu's Kalasalingam Academy of Research and Education
9920004293@klu.ac.in

M. Madhu Kiran
Computer Engineering and Science
Krishnankoil, Tamil Nadu's Kalasalingam Academy of Research and Education
9920004355@klu.ac.in

Abstract - Handwritten digit recognition is the ability of computers to recognise human-written numbers on paper. The machine has a challenging task since handwritten numbers can be produced in a variety of ways and aren't always accurate. The handwritten digit identification method, which recognises the digit that is present in the image using a photograph of a digit, is one solution to this issue. Using the MNIST dataset, we want to develop handwritten digit recognition in our project. Convolutional neural networks, a particular kind of deep neural networks, will be used. In the end, we'll develop a graphical user interface (GUI) that enables easy differentiation between a number and a drawing.

Keywords: GUI, machine learning algorithm, convolutional neural network, digit recognition from handwriting and the MNIST dataset.

I. INTRODUCTION

A well-known issue with image categorization is handwritten digit recognition. In this, we must group handwritten numerals into labels, ranging from 0 to 9. For this task, neural network models are a very effective and powerful classifier. Humans are intellectual creatures who can read and recognise the various handwritten characters and digits that their fellow beings have created. Using artificial intelligence and machine learning, we wish to imbue the same characteristics in a machine.

Many machine learning and deep learning approaches are being used by developers to make machines smarter. Applications for deep learning that use convolutional neural networks (CNN) include spam detection, object identification, face recognition, and picture classification. In addition to professional and commercial applications, handwritten digit recognition also has practical applications that can be highly beneficial to blind individuals. It also promotes the straightforward resolution of complex problems, simplifying our lives. For the purpose of recognising handwritten digits, many algorithms have been developed. However, they are still not up to standard because there are so many diverse writing styles. Another issue with the handwritten number recognition method is misclassification. result of poor contrast, picture text ambiguity, interrupted text stroke, undesired objects, distortion, disoriented patterns, as well as interclass and intraclass similarity.

Handwritten digit recognition (HDR) is a term used to describe a machine's capacity to recognise human-written digits. It is challenging job for a machine since handwritten digits are not always exact. This problem is resolved by our method, which identifies the digit represented by a picture of a digit. The purpose of this study is to analyse the outcomes of several handwriting algorithms. In our method, the CNN algorithm is trained using the Modified National Institute of Standards and Technology (MNIST) dataset. Computer gaming, data mining, handwriting identification, and computer vision applications have all benefited from using CNNs, a deep learning method.

II. DATASET

Yann LeCun, Corinna Cortes, and Christopher Burges developed the MNIST dataset, also known as the dataset altered by the National Institute of Standards and Technology. The fact that the size of the various scanned digits was normalised and their justification for being in the middle making it a suitable database for assessing these models. This made using machine learning to create handwritten digit recognition systems highly popular among academics. The error rate can be greatly decreased by utilising a variety of classifiers for various algorithms and parameters. The MNIST dataset includes a variety of handwritten digits as training examples. The training dataset was made up of 60,000 of the total 70,000 photos, while the testing dataset was made up of 10,000. The 10 numerals from 0 to 9 have been accurately marked on photos in both datasets using 28*28 grey scale graphics to represent handwritten numbers. Every MNIST data point consists of two components: an image of the handwritten digit itself and a target label pertaining to the handwritten digit. Utilising the MNIST dataset only requires minimal data cleaning, allowing one to concentrate on the goal of their deep learning model or machine learning.

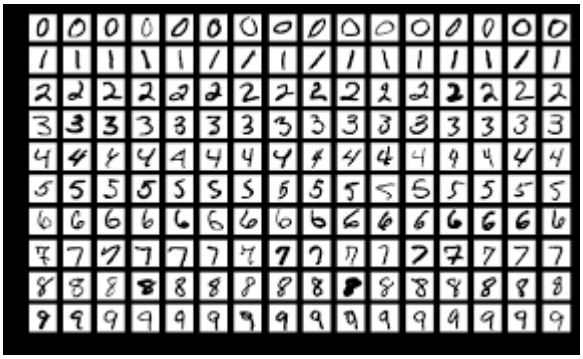


Fig 1: Various handwriting styles are used for the dataset's labels.

III. LITERATURE SURVEY

The 1980s saw the debut of the first handwriting recognition systems for numbers. There are several uses for classifier-assisted handwritten digit recognition, and it is crucial. Because they differ in size, thickness, and margin placement, numerals written by hand cannot be compared. These are some of the difficulties we encountered when looking for a solution. Y. LeCun et al. used back-propagation networks to recognise handwritten numbers.

X. Han et al. [1] summarise CNN's most recent developments, go into further depth on related image recognition technology research, and discuss CNN's application to handwritten number recognition. However, due to parallelism in digit form, every neural network has some mistake rate. R. Sudhakar et al. [2] A hybrid model for video super resolution was created by combining a deep learner driven by optimisation with a nonlinear regression model. Each low-resolution frame is obtained after framing by the non-linear regression model and the Fractional-Group Search Optimizer-based Deep Belief Network (FrGSO-DBN) classifier.

In order to remove extraneous information and preserve crucial elements, Caiyun Ma et al. [3] normalised the images of varied sizes and stroke widths. V.C. Bharathi et al. claim. [4], the index position for the keyword search is obtained by taking the important terms out of the text. They did this by calculating the keyword's Euclidean distance from the search terms. The most common problem was the resemblance of the numbers, such as 1 and 7, 5 and 6, 3 and 8, 9 and 8, etc. The same numeral can be written in a variety of ways as well.

Vineet Singh[5] and colleagues used PCA (principal component analysis) to recognise a digit using a single-layer neural network. The biggest disadvantage is that it takes more time to train. Retno Larasati et al. successfully used the approach of ensemble neural networks combined with ensemble decision trees despite their lower accuracy (84). To do handwritten digits recognition, they used ensemble neural networks and ensemble decision trees [6].

IV. PROPOSED SYSTEM

The digit recognition problem is a possible difficulty in the realm of recognition of handwriting. It's also one of the most challenging issues in computer vision and machine learning. It is referred to as a demanding undertaking since it is challenging to establish an accurate automatic recognition of handwritten digits. Using digital devices to fill out online forms and processing bank checks are two examples of digit recognition applications. Due to the variety of human handwriting, teaching computers to distinguish handwritten digits is difficult. This study[7] use machine learning approaches to address the difficulty of digit recognition. The goal of the project is to develop a tool that would allow for the analysis of various classification techniques and the selection of the most effective digit identification technique. This tool would comprise a number of different sorts of performance measures and categorization models.

Fig 1: The main steps of Using CNN, a Handwritten Digit Recognition System are illustrated. Four stages are included in the recommended approach for identifying and categorising the digits:

A. Pre-processing

B. Segmentation

C. Extraction of Features

D. Recognition and Classification

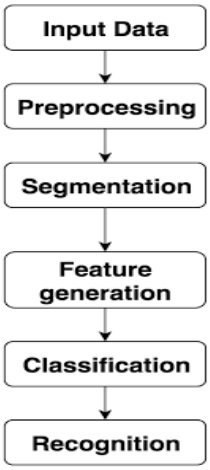


Fig 2: System Architecture

For problems with self-perception, such as understanding images, human voices, and robots moving around the environment, deep learning has emerged as a vital technique. Convolutional neural networks are a useful idea that we want to apply to digit recognition. It is advised to interpret CNN and use the handwritten digit recognition method with it. The 2D images are used to obtain the feature maps by the convolutional neural network. The feature maps might then be used to classify the images. Instead of having a layer of neurons that are totally coupled, the convolutional neural network evaluates the mapping of visual pixels with the neighbourhood space. Convolutional neural networks are effective methods for processing signals and images.

Neural networks with convolutions Deep artificial neural networks include convolutional neural networks. It may be used to group photos together based on similarities (photo search), classify them, and identify objects in scenes. It is capable of differentiating between faces, people, street signs, platypuses, and many other visual data elements. The fundamental component of CNN is the convolutional layer[8]. The parameters of the layer are learnable filters (also

known as kernels) that have a small receptive field but cover the whole depth of the input volume. Each filter is convolved across the width and height of the input volume during the forward pass, resulting in the computation of the dot product and the creation of a 2-dimensional activation map for that filter. As a result, the network gains knowledge about whether a certain type of feature is observed at a particular spatial point in the input. The down sampling layer, which is applied like convolutions one patch at a time, is then given the activation maps. The CNN algorithm also includes a fully connected layer that classifies output by assigning a label to each node.

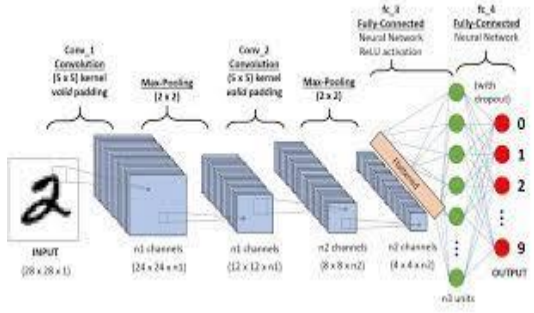


Fig 3: CNN Model

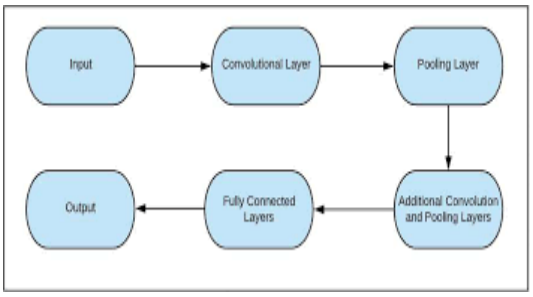


Fig 4: Model Architecture

a) Pre - Processing:

Pre-processing is in charge of performing various procedures on the provided picture. The image is fundamentally enhanced by being segmentation-friendly. To remove an interesting example from the backdrop is the primary driver behind pre-processing. The majority of noise filtering, smoothing, and standardisation should be completed at this point. A more condensed version of the example is characterised by the pre-processing[9]. A grayscale image becomes a binary image through the process of binarization. The initial method for the training set photos is to threshold them into a binary image, which will reduce the amount of data. Data normalisation is the initial stage in the preparation of data. Applying distance computations to it is done in this manner. In order to do this, the data must be transformed to fall into a more constrained or typical range, like [0, 1]. The raw picture data is recorded as an 8-bit unsigned integer with a high value range of [0, 255] for each pixel (attribute). A smaller range for an attribute will result from its expression in smaller units, which tends to give such attributes more "weight" or impact.

After normalisation, noise was reduced using the median filter. Specifically Gaussian noise is being removed from the image using this nonlinear digital filtering technique. Since the edge is a crucial component of a picture, we utilised the median filter since it protects the edge while reducing noise.

After preprocessing, the image sharpening process creates a mask of the original picture using a blurred, or "unsharp," negative image. A crisper picture than the original is then produced by combining the unshaped mask with the positive (original) image. A filter used in sharpening boosts the high-frequency parts of a signal. Following the Median Filter is an essential step because the Median Filter weakens the entire image in addition to removing noise. Some of the useful information that Median Filter has diminished can be improved or restored by sharpening.

Utilising attribute reduction techniques, one may get a considerably smaller, more compact representation of the original data set while closely maintaining its integrity. To put it another way, mining on the smaller data set ought to be more fruitful while still producing roughly the same analytical results. There are 784 properties in all for each original picture. It would be advantageous to minimise the total characteristics to a manageable number, making the data more efficient and processing simpler.

By using attribute reduction techniques, The original data's accuracy can be carefully preserved while producing a much smaller, condensed version of the data set. In other words, mining the smaller data set ought to be more effective while still yielding about equivalent analytical outcomes. A total of 784 properties are present in each original image. To make the data more efficient and easier to process, it would be advantageous to limit the total attributes to a manageable number.

b) Segmentation:

Following pre-processing of the input photographs, a series of images is employed to produce sub-images of distinct digits. A sub-image of separate digits with a unique number is created from pre-processed digit pictures. Each digit's size is translated into pixels. In this stage, the pictures from the dataset are segmented using an edge detection algorithm. There are little representations of each number within a larger image of a succession of numbers. An input image that has been previously processed is divided into isolated digits by employing a labelling procedure to give each digit a number.

c) Extraction of Features:

The pre-processed images are then represented as a matrix made up of pixels from the extremely large-sized images after the segmentation and pre-processing processes have been finished. The digits in the photographs that contain the required data will be advantageously represented in this way. Features are extracted during this process. Redundancy in the data is eliminated at the feature extraction stage.

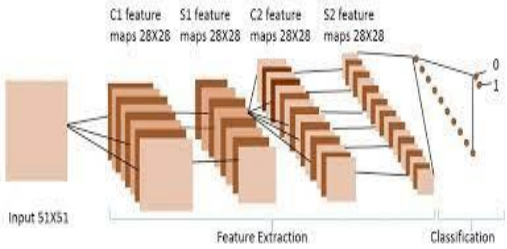


Fig 5: Extraction of Features

The output layer, feature extraction, and feature dimension transposition are the three parts of the architecture this study proposes[10]. Dense blocks are used by the author to carry out feature extraction using Dense Net. Each input picture output layer is automatically extracted by these dense blocks into a feature sequence. The loss is then calculated during the training process using this feature sequence, and the prediction results are produced during the testing step.

d) Recognition and Classification*:*

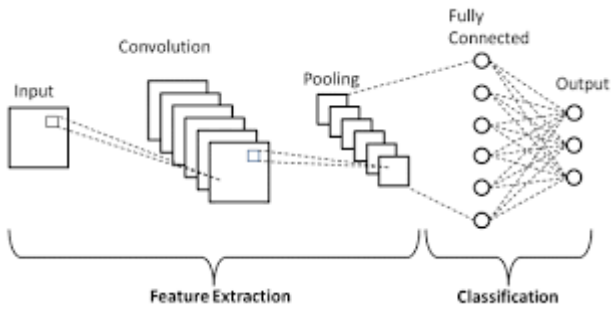


Fig 6: A simple structure of CNN

Following the Feature Extraction phase, the individual Deep neural networks(DNN), convolutional neural networks(CNN), and deep belief networks algorithms are trained using the training pictures. Identification and Classification: Following the training phase, " The decision-making stage of a recognition system is the categorization and recognition phase and it utilises the characteristics collected in the prior phase. The data are categorised using a feed forward back propagation neural network with two hidden layers and an architecture of 54 - 100 - 100 - 38[11]. Due to the requirement that one of the digits be identified, the output layer is a layer that competes, however the hidden layers employ the log sigmoid activation function.

V. METHODOLOGY

1. Load the dataset and import the libraries

All of the components that we require to first train our model will be imported. One of the datasets that is already a part of the Keras package is MNIST. As a result, utilising and importing the dataset are straightforward procedures. The mnist.load_data() function gives us access to both the training and testing sets of data and labels.

2. Data preparation

We must first carry out specific operations and assess the data before supplying it to the neural network; otherwise, the input cannot be delivered straight to the network. The practise data's dimension is (60000,28,28). Given that the CNN model requires an extra dimension, we rebuild the matrix to (60000,28,28,1).

3. Create the prototype

The following stage will include creating our CNN model in Python. Convolutional and pooling layers make up the majority of a CNN model. Because it performs better with data that is arranged as grid structures, CNN performs well when it comes to jobs involving the categorization of pictures. By using the dropout layer to block certain neurons, the model's offer fitting performance declines throughout training. The Adadelta optimizer will then be used to put the model together.

4. Train the prototype

With the aid of Keras' model, the model will begin to be trained.fit() method. It's vital to have the required batch size, epochs, training data, and validation data. The model's training process takes some time. We store the weights and model specifications in the "mnist.h5" file after training.

5. Evaluate the prototype

To gauge how well our model performs, 10,000 photos from our dataset will be utilised. As a result, the testing data represents new data for our model because it was not used to train the model. We can attain 99% accuracy since the MNIST dataset is evenly distributed.

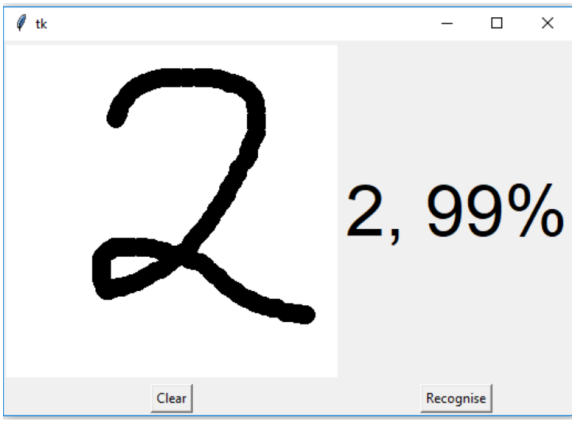
6. Create GUI to predict digits

In a new file that we've created, we've created an interactive window providing the graphical user interface that enables us to create numbers on a canvas and then use buttons to identify those values. The Tkinter library is a part of the default Python library. The trained model is used by the predict_digit() function we created to predict the digit from the input picture.

VI. RESULTS AND DISCUSSIONS

This section contains the findings and comments for the CNN-based handwritten digit recognition system. Also included is a working example of the handwritten digit recognition method using CNN.

The programme class, which is in charge of creating the GUI for our programme, was the next thing we created. By recording the mouse event, we create a canvas that can be used for drawing. Pressing a button causes the predict_digit() method to be called, and the results are then shown.



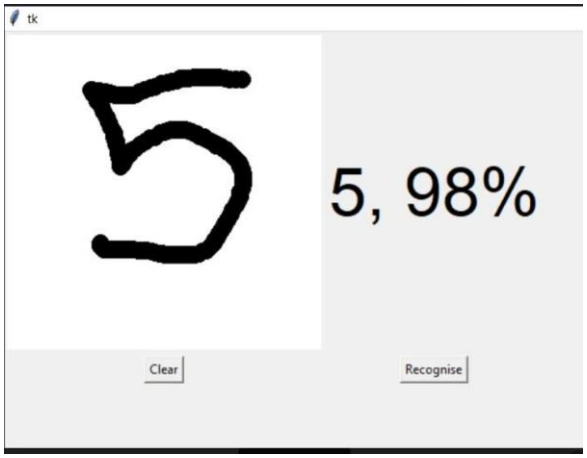


Fig 7: Recognizing the digits on GUI.

When we draw the digit on the GUI it gives the predicted digit along with accuracy of the match so that we can easily find the digit to be recognized. The GUI also have the options as clear to clear the canvas and also recognise button to recognize the digit.

VII. REFERENCES

[1] Xiaofeng Han and Yan Li (2020), “The Application of Convolution Neural Networks in Handwritten Numeral Recognition” in International Journal of Database Theory and Application, Vol. 8, No. 3, pp. 367-376

[2] M. Wu and Z. Zhang, "Handwritten digit classification using the mnist data set," Course project CSE802: Pattern Classification & Analysis, 2021.

[3] Caiyun Ma, Hong Zhang (2020), “Effective Handwritten Digit Recognition Based on Multi-feature Extraction and Deep Analysis”, 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 297-301

[4] V.C.Bharati, K.Veningston, and P.V.Venkateswara Rao, “Query -Based Word Spotting in Handwritten Documents Using HMM", Springer Nature Singapore pte Ltd.2020,Data Engineering and communication Technology, Advances in Intelligent systems and computing 10789.

[5] Vineet Singh, Sunil Pranit Lal, “Digits recognition using single layer neural Network with principal component analysis”, Computer Science and Engineering (APWC on CSE), 2019 Asia-Pacific World Congress IEEE, 4-5 Nov 2019.

[6] Retno Larasati, Hak Keung Lam, “Handwritten digits recognition using ensemble neural networks and ensemble decision tree”, Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), 2019 International Conference on, 8- 10 Nov. 2019.

[7] L. Bottou, C. Cortes, “Comparison of Classifier methods a case study in handwritten digit recognition”, Pattern Recognition,1994. Vol. 2 Conference B; Image Processing, Proceedings of the 12th IAPR International. Conference IEEE, 06 August 2020.

[8] Swapna Prava Ekka, “Recognition of Handwritten Digits using Proximal Support Vector Machine”, (2019), National Institute of Technology Rourkela, unpublished.

[9] 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.

[10] Pranit Patil, Bhupinder Kaur, “Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models”, Journal of Innovative Research in Computer Science & Technology (IJIRCST), Vol. 8, Issue- 4, July (2020), ISSN: 2347-5552.

[11] Y. LeCun et al., "Handwritten digit recognition with a backpropagation network," in Advances in neural information processing systems, 1990, pp. 396-404