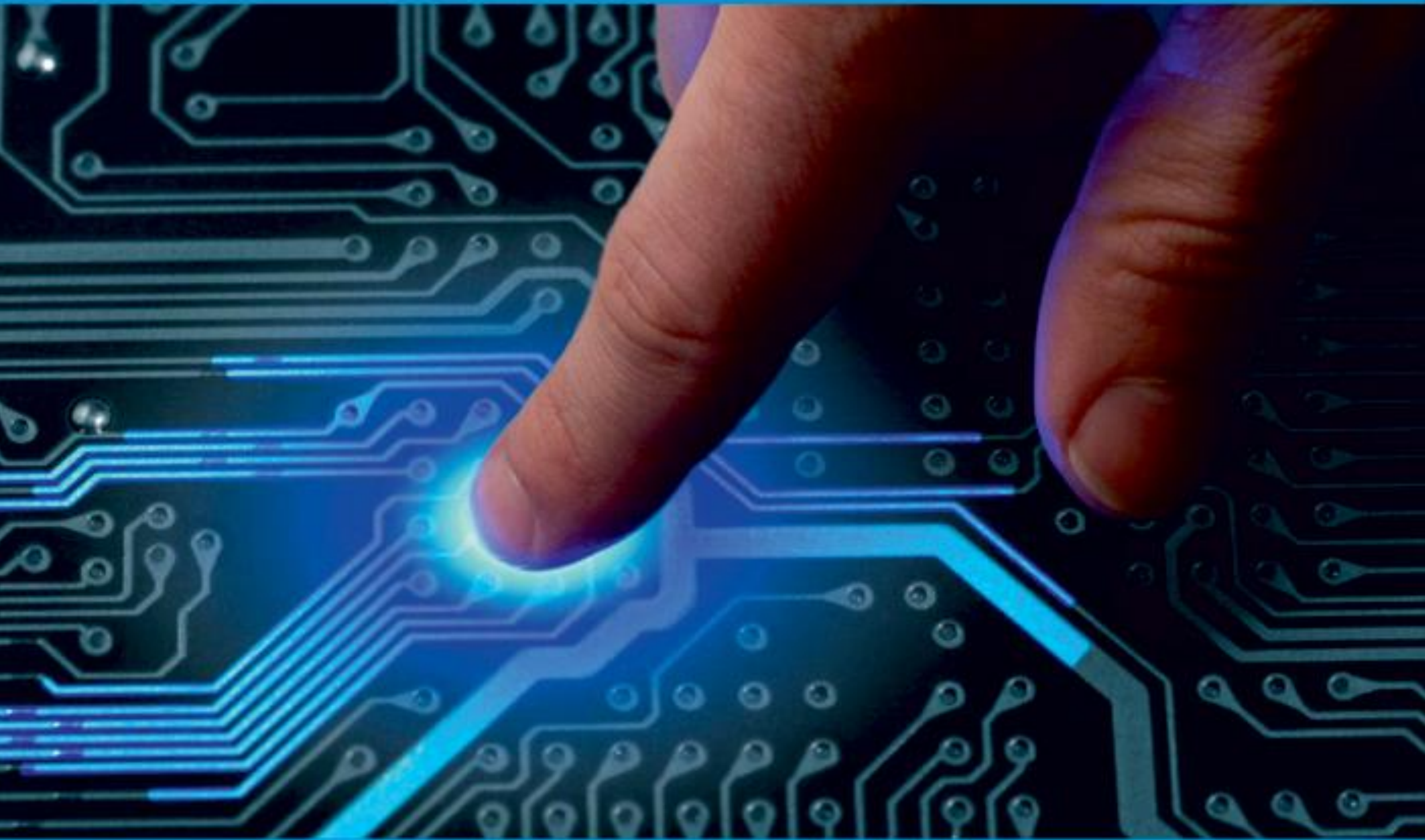
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Special Issue 1, February 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

9940 572 462 6381 907 438 ijircce@gmail.com www.ijircce.com

A Novel Method for Handwritten Digit Recognition

Mrs. V. Sundara Jeyalakshmi[1], N. Hamsavashini[2], V. Harshitha[3], M. Hemavathi[4], D. Pavithra[5], S. Pradeepa[6]

Assistant Professor, Department of Electronics and Communication Engineering, Adhiyamaan College of Engineering, Hosur, Krishnagiri, Tamil Nādu, India. [1]

UG Scholars, Department of Electronics and Communication Engineering, Adhiyamaan College of Engineering, Hosur, Krishnagiri, Tamil Nādu, India. [2,3,4,5,6]

ABSTRACT - Many of us still find it challenging to read handwritten material since everyone of us has a unique way of understanding it. As the world is moving towards digitization, converting the handwritten information to a readable digital format reduces the difficulty and the human effort. With the help of machine learning and deep learning algorithms, the handwritten patterns can be recognized and classify them accordingly to a digital format with human level accuracy. The major goal of this research is to offer effective and trustworthy methods for handwritten digit recognition. To implement this, we will use a special type of deep neural network called the Convolutional Neural Network (CNN) from Machine learning. Using the MNIST (Modified National Institute of Standards and Technologies) database and compiling with the CNN gives the basic structure of the project development. The MNIST data set includes over 60,000 training images for handwritten numbers from 0 to 9 and over 10,000 test images. To perform this model we need some libraries such as NumPy, Pandas, Tensorflow, Keras.

KEYWORDS: CNN, MNIST, Tensorflow, Keras, NumPy, Pandas.

I. INTRODUCTION

Machine learning and deep learning plays an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, prediction and many more areas. This article compares classifiers like KNN, PSVM, NN, and Convolutional Neural Network (CNN) on the basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with The classifiers to recognise the handwritten digits (0–9) from the well-known MNIST dataset. To make machines more intelligent, the developers are diving into machine learning and deep learning techniques. Different neural network topologies are used by deep learning for various types of tasks. For instance, object detection, image segmentation, object recognition, and sound and image categorization. The capacity of computers to detect human handwritten digits is known as handwritten digit recognition. Because handwritten numerals are imperfect and can be generated with a variety of tastes, it is a difficult work for the machine. Handwritten digits can be categorised and recognised using deep learning and machine learning algorithms.

II. RELATED WORK

The results of some of the most widely used machine learning algorithms, such as SVM, KNN, and RFC, as well as deep learning calculations, such as multi-layer CNN using Keras with Theano and Tensorflow. These were used to produce an accuracy of 98.70% using CNN (Keras + Theano) as opposed to 97.91% using SVM, 96.67% using KNN, and 96.89% using RFC. Comparatively, CNN is the most effective 5 solution for all types of prediction issues, including those using picture data. Next, by comparing the execution times of the algorithms, they determined that increasing the number of epochs without changing the configuration of the algorithm is pointless due to the limitation of a certain model, and they discovered that beyond a certain number of epochs, the model begins over- fitting the dataset and provides biased predictions. Rectified linear units (ReLU) activation and a convolutional neural network (CNN) that incorporates the Deep learning 4j (DL4J) architecture is used to recognize handwritten digits. Two techniques are previously used namely Pattern Recognition and Artificial Neural Network (ANN). Both techniques are defined and different methods for each technique is also discussed. Methods for pattern recognition include Bayesian

Decision Theory, Nearest Neighbor Rule, and Linear Classification or Discrimination. Neural networks are used for shape identification, Chinese character recognition, and handwritten digit recognition. Written digits are trained and identified using neural networks. Following testing and training, the accuracy rate was 99%.

III. METHODOLOGY

We used MNIST as a primary dataset to train the model, and it consists of 70,000 handwritten raster images from 250 different sources out of which 60,000 are used for training, and the rest are used for training validation. Our proposed method mainly separated into stages, preprocessing, Model Construction, Training & Validation, Model evaluation & Prediction. Since the loading dataset is necessary for any process, all the steps come after it. Libraries used in this project are Keras, Tensor flow, Numpy, Tkinter.

Loading The Data Set Mnist Data Set

Modified National Institute of Standards and Technology (MNIST) is a large set of computer vision dataset which is extensively used for training and testing different systems. The database contains 60,000 images used for training as well as few of them can be used for cross validation purposes and 10,000 images used for testing. All the digits are grayscale and positioned in a fixed size where the intensity lies at the center of the image with 28×28 pixels. It forms an array which can be flattened into 28*28=784 dimensional vector. Each component of the vector is a binary value which describes the intensity of the pixel.

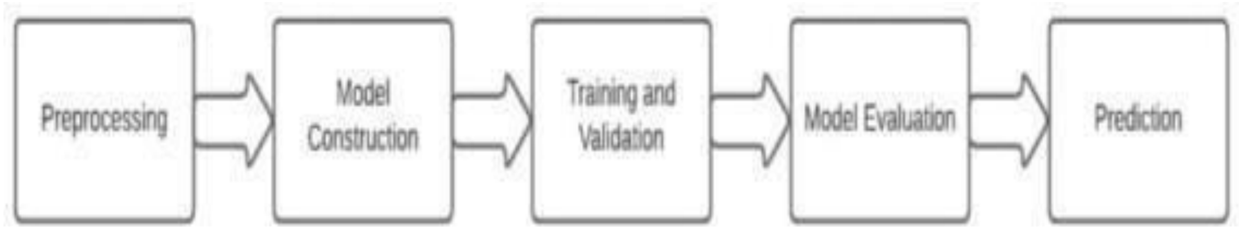


Fig 1: Steps for recognition system

Pre-Processing

Data pre-processing is a data mining technique which is used to shape the input images in a form suitable for segmentation pre-processing is used. Data pre-processing is a necessary step before building a model with these features. It usually happens in stages.

- **Data quality assessment** - helps to identify those records that have become inaccurate
- **Data cleaning** - if we have a well-cleaned dataset, we can get desired results even with a very simple algorithm.
- **Data transformation** -turning the data into an appropriate format for the computer to learn from.

Data Reduction Techniques

Data reduction can be used to reduce the amount of data and decrease the costs of analysis. After loading the data, we separated the data into X and y where X is the image, and y is the label corresponding to X. The first layer/input layer for our model is convolution. We must rearrange the pictures such that each pixel value is in its own space since convolution treats each pixel as a neuron. This transforms a 28x28 matrix of greyscale values into a 28x28x1 tensor. We may divide the photos into train and test for next stages whether they are all the correct dimensions.

Data Encoding

Keeping categorical variables in their current form would be problematic since machine learning is dependent on mathematical equations. Although many algorithms naturally allow categorical values, the question of whether or not to encode the variables in certain circumstances is still up for debate. The next stage is to process the data in a form that is suitable to feed into machine learning models after the data types and characteristics included in the data set have been identified. which is recognized during validation. The three popular techniques of converting Categorical values to Numeric values are done in two different methods.

- Label Encoding.
- One Hot Encoding.
- Binary Encoding.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 11, Special Issue 1, February 2023 ||

The term "encoding variability" refers to the diversity in individual encoding within a category. When we discuss the variability of a single hot encoding, the variability is dependent on the point in implementation at which it chooses the number of categories to include that have an adequate influence on the objective. Other encoding techniques do exhibit a substantial degree of variability, which is seen during validation.

CNN Layer:
A CNN typically consists of a number of convolutional layers, but it also includes additional elements. Remember that the higher convolution layers recognize complicated objects; the classification layer in the final layer of a CNN uses the output of the final convolution layer as input. The classification layer generates a series of confidence ratings (numbers between 0 and 1) that indicate how probable it is for the picture to be a member of a "class," based on the activation map of the final convolution layer. The following steps are taken after choosing the model: We'll be utilizing the Sequential model type. The simplest technique to create a model in Keras is sequential. It enables layer-by-layer model construction. To add layers to our model, we employ the "add()" method. We have Conv2D layers as our initial two layers. These layers of convolution will work with our input pictures, which are represented as two-dimensional matrices. There are 32 nodes in the second layer and 64 in the first layer respectively. Depending on the size of the dataset, this amount can be changed to be greater or lower.

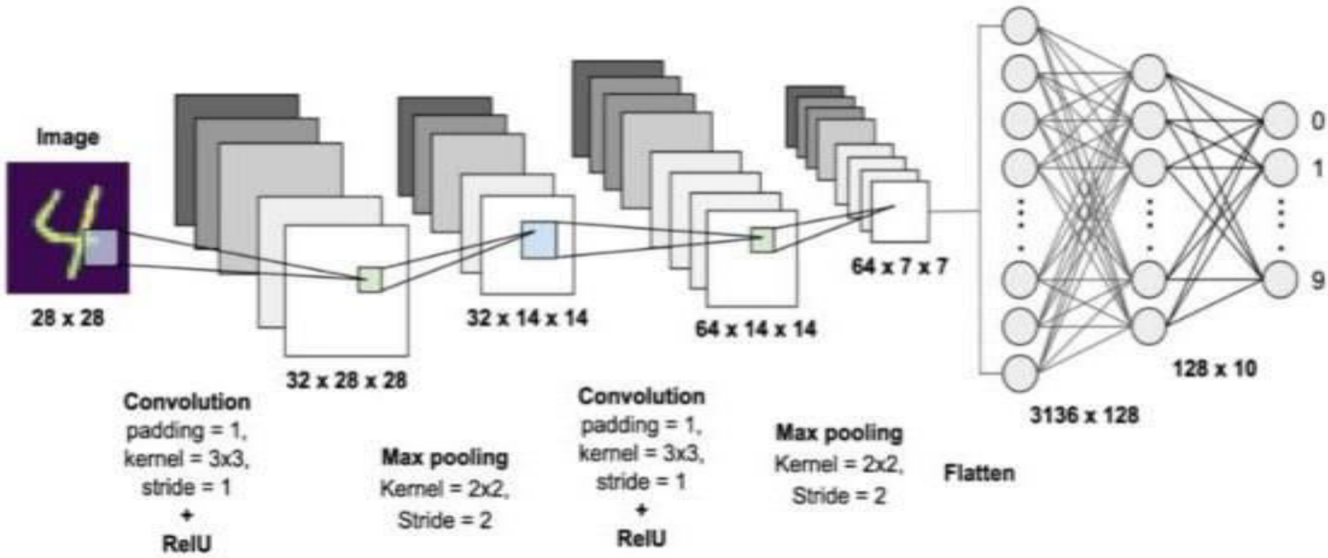


Fig 2 :CNN Architecture

Rectified linear unit [ReLU] layer:
Given activation function on the image-capturing data. When back propagation occurs, the ReLU function is utilized to keep the values of the pixels from changing
.

Pooling layer: Performs a down-sampling operation in volume along the dimensions (width, height).

Fully connected layer: score class is focused, and a maximum score of the input digits is found. There is a significant rise in complexity as we dig further and deeper into the levels. However, it would be worthwhile to go since while accuracy might improve, time consumption does, sadly, too.

Working of CNN:
Convolutional neural networks are made up of several artificial neuronal layers that add up all of the inputs' weights to get an activation value. The artificial neurons of a CNN can recognize many visual aspects when supplied with pixel values. Each layer in a ConvNet creates a number of activation maps when you feed it an image. Activation maps draw attention to the important aspects of the image. As input, each neuron multiplies a patch of pixels' color values by its weights, adds them all together, and then sends the result through the activation function.

Typically, the first (or bottom) layer of a CNN recognizes fundamental characteristics such edges on the horizontal, vertical, and diagonal axes. The second layer extracts more complicated characteristics, such as corners and combinations of edges, using the output of the first layer as its input. Convolutional neural networks have layers that recognize higher-level elements like objects, faces, and other features as you go deeper into the network. Convolution is the process of multiplying weights by pixel values and adding them together.

A CNN typically consists of a number of convolutional layers, but it also includes additional elements. Remember that the higher convolution layers recognize complicated objects; the classification layer in the final layer of a CNN uses the output of the final convolution layer as input. The classification layer generates a series of confidence ratings (numbers between 0 and 1) that indicate how probable it is for the picture to be a member of a "class," based on the activation map of the final convolution layer. The following steps are taken after choosing the model: We'll be utilizing the Sequential model type. The simplest technique to create a model in Keras is sequential. It enables layer-by-layer model construction. To add layers to our model, we employ the "add()" method. We have Conv2D layers as our initial two layers. These layers of convolution will work with our input pictures, which are represented as two-dimensional matrices.

There are 32 nodes in the second layer and 64 in the first layer respectively. Depending on the size of the dataset, this amount can be changed to be greater or lower. In our situation, 64 and 32 function nicely, therefore we will continue using this. The filter matrix for our convolution has a kernel size. We will thus have a 3x3 filter matrix if the kernel size is 3. The Rectified Linear Activation, often known as ReLU, will serve as the activation function for our first two layers.

In neural networks, this activation function has been shown to function effectively. Our first layer also accepts a shape as an input. Each input picture has the following shape: 28,28,1, where the 1 denotes that the photos are in grayscale. There is a "Flatten" layer sandwiched between the Conv2D layers and the dense layer. Between thick layers and convolution, flatten acts as a link. Our output layer will be of the layer type "Dense."

After the model is built, it must be assembled in order to train it using the provided data set. The model is assembled with optimizers. The learning rate is managed by the optimizer. "Adam" will be our optimizer in this case. We will use the "accuracy" measure to check the accuracy score on the validation set when we train the model to make things even simpler to understand. The goal of training and testing any data model is to maximize validation and learning rates. For real-world image classification prediction, we need to do a little image pre-processing. The steps of image pre-processing are

1. Loading image
2. Convert the image to greyscale
3. Resize the image to 28x28
4. Converting the image into a matrix form
5. Reshape the matrix into 28x28x1

After pre-processing, we predict the label of the image by passing it through the neural network. The output we get is a list of 10 activation values 0 to 9, respectively. The position having the highest value is the predicted label for the image.

IV. RESULT

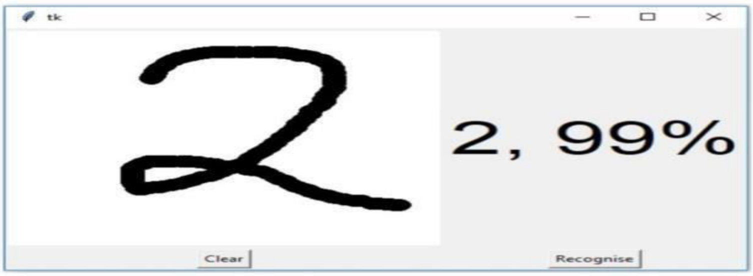


Fig 3: output sample ‘digit 2’

Thus the handwritten digit 2 is recognized with the accuracy of 99%.

Fig 4: output sample 'digit 5'

Thus the handwritten digit 5 is recognized with the accuracy of 70%.

V. CONCLUSION

Our project handwritten digit recognition deals with identifying the digits. The main purpose of this project is to build an automatic handwritten digit recognition method for the recognition of handwritten digit strings . In this project, different machine learning methods, which are SVM (Support vector Machine), ANN (Artificial Neural Network) and CNN (Convolutional Neural Network) architectures are used to achieve high performance on the digit string recognition problem.

The proposed system takes 28 x 28 pixel sized images as input. The same system with further modifications and improvements in the dataset and the model can be used to build Handwritten Character recognition System which recognizes human handwritten characters and predicts the output.

REFERENCES

1. Md,Anwar Hossain Md, Mohon AI, Recognition of handwritten digit using Convolutional neural network(CNN), IEEE, vol, 19, no, 2,2019
2. Y. Yin.J.Wu, and H.Zheng, 'Nefm: Accurate handwritten digits using convolutional neural networks', in 2016 International joint conference on neural netwrks (UCNN),2016, pp.525-531:IEEE.
3. Yousssoufchherawala, Parthapratim Roy and Mohamed cheriet, 'Feature set evaluation for offline handwriting recognition system: application to recurrent neural network', IEEE tranactions on cybernetics, vol.46
4. Ankit sharma , YogirajBarole, Kaustubh Kerhalkar, Neural network based handwritten digit recognition for Managing Examination Score in Paper Based test, IEEE,vol 5

BIOGRAPHY

Mrs. V. Sundara Jeyalakshmi. M.E,
Assistant professor/Department of ECE,
Adhiyamaan College of Engineering,
Hosur–635130.

N. Hamsavashini,
Bachelor of Engineering(student),
Adhiyamaan College Of Engineering, Anna University

V. Harshitha,
Bachelor of Engineering(student),
Adhiyamaan College Of Engineering,
Anna University

M. Hemavathi
Bachelor of Engineering(student),
Adhiyamaan College Of Engineering,
Anna University

D. Pavithra,
Bachelor of Engineering(student),
Adhiyamaan College Of Engineering,
Anna University

S. Pradeepa,
Bachelor of Engineering(student),
Adhiyamaan College of Engineering, Anna University