IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HANDWRITTEN DIGIT RECOGNITION USING PYTHON

[1]Penan Rajput, [2]Jayesh Nahar, [3]Sanjeev Kumar, [4]Aniket Pathak

[1], [2, 3] Final Year B.Tech. Student, [4] Assistant Professor

[1,2,3,4] Department of Computer Science and Engineering,

[1,2,3,4] Hindi Seva Mandal's Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal, Maharashtra, India

Abstract: Handwritten digit recognition is the intelligence of computers to recognize digits written by humans. But it becomes one of the most challenging tasks for machines as handwritten digits are not perfect and can be made with many different: flavors, size, thickness. Thus, as a solution to this problem, Handwriting digit recognition model comes into picture. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. Among the three famous NN approaches: deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN), the specialization of CNN as compared to other NN of being able to detect pattern is what makes it so useful for recognizing handwritten digits in this paper. Our goal is to implement a CNN based handwritten digit recognition model that uses the image of a digit and recognizes the digit present in the image. The performance is tested on MNIST dataset. The network was trained on 60,000 and tested on 10,000 numeral samples. We carried out extensive experiments and achieved a recognition accuracy of 99.87%.

Index Terms - Handwritten digit recognition, Machine learning, Deep Learning, Convolutional neural network, digit recognition, classification

1. INTRODUCTION

Humans can very easily see, read & write any handwritten digits, when written in proper format. Even if the digits are not written in proper format we can use our logic and predict what digit it could be. But It is a hard task for the machine to recognize handwritten digits as these are not perfect and can be made with many different flavors. Thus handwritten digit recognition is the solution to this problem which uses the image of a digit and recognizes the digit present in the image.The handwritten digit recognition is the ability of computers to recognize human handwritten digits. Since the 1980s handwriting recognition of digits has been around. Research in this area has been a standard area because of its usage in technologies such as postal mail sorting, bank check processing, form data entry, etc. Even though any digital image is a matrix of 0's and 1's, the computer comes to know whether the input image is a digit by recognizing some specific formats. Thus a handwritten digit recognition model helps in recognizing such patterns which in turn recognizes the digits. Our goal is to build a model that can efficiently and reliably recognize the digits and output the proper result. Amongst all the other neural networks, working and implementing a model using Convolution Neural Network gives out the most precise results. It is most popularly used for analyzing images as well as for other data analysis or classification problems. CNN has hidden layers called convolutional layers. These layers work the same way as other layers do but here we need to specify the no of filters each layer should have . These filters are actually what detects the pattern. Patterns could be edges, corners, circles or any complex other objects like eyes, ears or even deeper full dogs, cats, etc. Thus, the specialization of CNN as compared to other NN of being able to detect patterns is what makes it so useful for recognizing handwritten digits.

2. LITERATURE REVIEW

In the paper, "Handwritten Digit Recognition Using Deep Learning" the authors, "Anuj Dutt, Aashi Dutt" have compared the results of some of the most widely used Machine Learning Algorithms like KNN & RFC and with Deep Learning algorithm like multilayer CNN using Keras with Theano and Tensorflow. Using these, they were able to get the accuracy of 98.70% using CNN (Keras+Theano) as compared to 97.91% using SVM, 96.67% using KNN, 96.89% using RFC [1].

The authors of the paper, "Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning", had applied neural network methods to a large, real-world task. Our results appear to be the state of the art in digit recognition. We demonstrated that a general-purpose neural network chip can be incorporated as an accelerator in a large network. They found that real problems with regularity scale well. They also showed that a network can be trained on a low-level representation of data that has minimal preprocessing [2].

In the paper, "Unconstrained Handwritten Numeral Recognition Using Majority Voting Classifier" the authors, "Rajiv Kumar, Pervez Ahmed, Mayank Kumar Goyal, Amresh Kumar" presented a simple profile, combined local & global features and majority voting scheme classifier for unconstrained handwritten numeral recognition. Linear discriminant analysis and KNN classifiers are used for classifying these features. A A majority voting scheme has been performed with three neural network classifiers and KNN classifiers. The performance is tested on MNIST dataset. The network was trained on 60,000 and tested on 10,000 numeral samples of which 98.05 % test samples are correctly recognized [3][4].

The authors, "Stefan Knerr, LCon Personnaz, and GCrard Dreyfus" of "Handwritten Digit Recognition by Neural Networks with Single-Layer Training" had introduced the STEPNET procedure, which decomposes the handwritten digit recognition problem into simpler subproblems that can be solved by linear separators. They presented results from two different databases:a European database comprising 8700 isolated digits, and a zip code database from the U.S. Postal Service comprising 9000 segmented digits [5].

3. METHODOLOGY

In this part we are depicting the different advances and acknowledges like general algorithm, strategies, datasets utilized, how the models are made and how the models were prepared are tried.

3.1. GENERAL ALGORITHM

The Figure 3.1 portrays the Data flow diagram of the proposed system model.

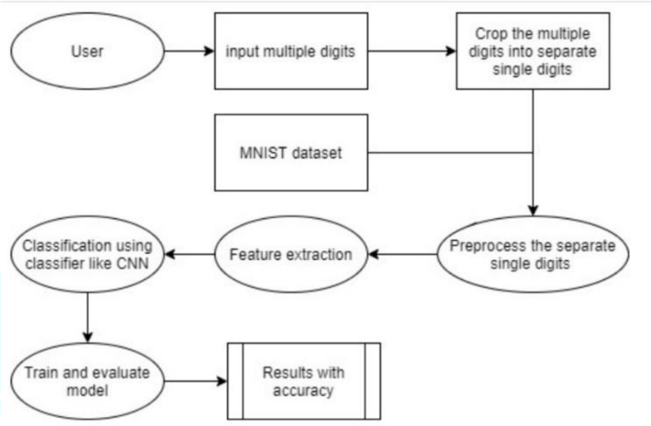


Fig. 3.1 Plotting of Some Random MNIST Handwritten Digits

The steps shown in the figure are as follows :

1. **input multiple digits** in this step, user provides the image of digits by drawing image on screen.
2. **crop the image into separate single digits** for situation draws the multiple images on screen and crop the digits into separate single digits
3. **MNIST dataset** for providing the dataset to further steps
4. **image preprocessing** which converts the color image to a monochrome one, prepared for further analysis;
5. **feature extraction** involves reducing the number of resources required to describe a large set of data
6. **classification using CNN** is the stage in which the value of the digit is computed with a completely different algorithm.
7. **train and evaluate model** The train-test split is a procedure for assessing the performance of an machine learning algorithm
8. **Results and accuracy** returns the most probable value of the digit with a confidence. We cannot rely solely on this results, therefore the next step is required;

3.2. MNIST DATASET USED

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image [6]. Modified National Institute of Standards and Technology (MNIST)[7] is a large set of computer vision dataset which isextensively used for training and testing different systems. It was created from the two special datasets of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits from 250 people, among them 50% training dataset was employees from the Census Bureau and the rest of it was from high school students [8]. However, it is often attributed as the first datasets among other datasets to prove the effectiveness of the neural networks. Figure 3.2 shows the Distribution of Each Digits for Training and Testing.

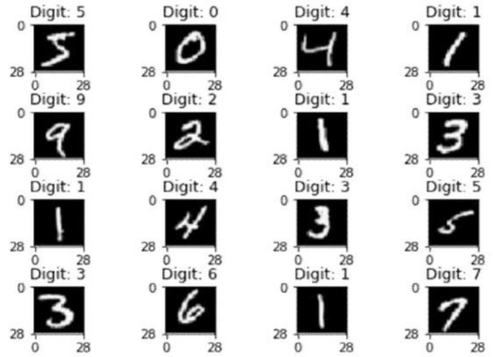


Fig. 3.2 Dataflow Diagram of General Algorithm

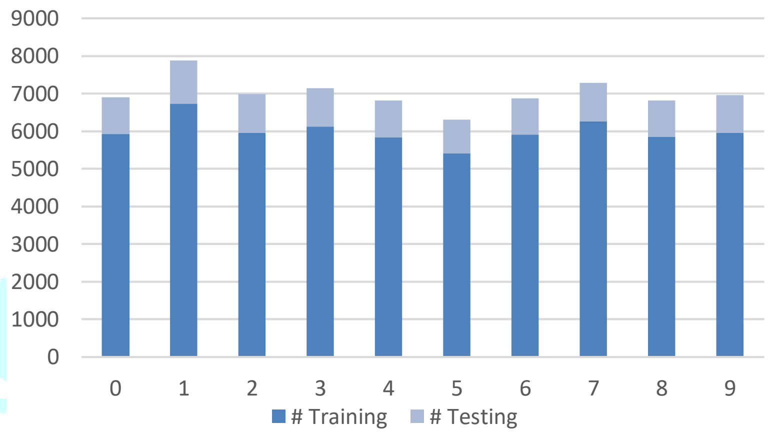


Fig. 3.3 Distribution of Each Digits for Training and Testing

The database contains 60,000 pictures utilized for preparing for cross-validation purposes and 10,000 pictures utilized for testing [9]. All the digits are gray scale and positioned in a fixed size where the intensity lies at the center of the image with 28×28 pixels. Since all the images are 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 [10]. Figure 3.3 shows the some of the digits from MNIST dataset.

3.3. CONVOLUTIONAL NEURAL NETWORK

CNN is a deep learning algorithm that is widely used for image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It inputs the image in the form of small chunks rather than inputting a single pixel at a time, so the network can detect uncertain patterns (edges) in the image more efficiently. CNN contains 3 layers namely, an input layer, an output layer, and multiple hidden layers which include Convolutional layers, Pooling layers (Max and Average pooling), Fully connected layers (FC), and normalization layers [11]. CNN uses a filter (kernel) which is an array of weights to extract features from the input image. CNN employs different activation functions at each layer to add some non-linearity [12] [13].

4. IMPLEMENTATION

4.1. PROPOSED ALGORITHM

Algorithm for handwritten written digit recognition generally comprise of three stages.

- **Step 1** Pre-processing. This progression should eliminate immaterial information and information that could have negative impact on acknowledgment. Common steps in this fluster are binarization, standardization, smoothing, denoising
- **Step 2** Feature extraction. Feature extraction involves reducing the number of resources required to describe a large set of data.
- **Step 3** Classification. Classification is a process related to categorization, the process in which ideas and objects are recognized, differentiated and understood.

4.2. PREPROCESSING

Pre-handling is an underlying advance in the machine and profound realizing which centers around working on the information by reducing undesirable pollutants and repetition. To rearrange and separate the information every one of the pictures present have been reformed in 2-dimensional pictures i.e. (28,28,1). Every pixel esteem of the pictures lies between 0 to 255 followed by Normalizing these pixel esteems by changing over the dataset into 'float32' and then, at that point isolating by 255.0 so the info highlights will go between 0.0 to 1.0. Then, one-hot is performed encoding to convert the y esteems into zeros and ones, making each number straight out, for instance, a yield esteem 4 will be changed over into a variety of nothing and one i.e. [0,0,0,0,1,0,0,0,0,0] [13]. The accuracy in recognition of handwritten numerals can be improved by preprocessing the raw data. With a brief investigation of the raw image data, the main issues discovered are image noise and unrecognizable handwriting. Thus, preprocessing of the raw data is deemed to be necessary before training them. A series of image processing techniques are conducted to lead through the preprocessing stage.

4.3. NOISE REDUCTION

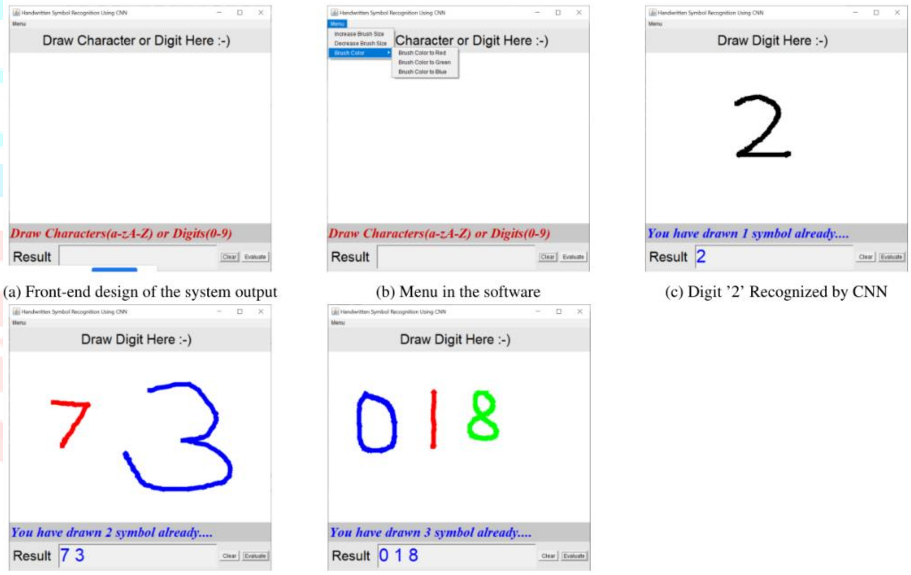
The idea of image noise reduction is to train a model with noisy data as the inputs, and their respective clear data the outputs[14]. Noise reduction is the process of removing noise from a signal. Noise reduction techniques exist for audio and images. Noise reduction algorithms may distort the signal to some degree [15].

4.4. EXPERIMENT

So now let's dive into the implementation of our project. In practical result of our software. We have designed a simple and very user-friendly UI for own software.

The Figure. 4.1a shows the front-end design of the system output. The following figures show the sequence of steps to be carried out to obtain the required output.

1. It has a menu button at the top, that given a drop-down list of" Increase Brush Size"," Decrease Brush Size"," Brush Color" options on click. Figure. 4.1b shows the functionality of software.
2. The center panel of our software consists of the area where users can actually draw their favorite digit.
3. The Figure 4.1c depicts how user draws shape of digit 2 on empty space with help of mouse pointer. Our software is locally capable of recognition digit (0-9).
4. After drawn the digit inside box, the bottom panel of on" Evaluate" button which will process the digit drawn by the user and will recognize and deploy the digit on the result and digit (0-9) of us like simple. We need to move our cursor to the center panel and drawn any digit in whichever form want and the software will undoubtedly process the input given to it and deploy the correct output in the correct form. The example is shows in Figure 4.1c.
5. User can actually draw multiple digits on screen and lift the mouse during the drawing digit and draw another digit if wants to. The Fig. 4.1d shows how digits '7', '3' are drawn on screen and output generated digits '7', '3' matching with input.
6. Fig. 4.1e shows how digits '0', '1', '8' is drawn by user and output evaluated correctly by CNN classifier. The bottom panel has an" Evaluate" button that on clicking processing the input and then shows the result. The



(a) Front-end design of the system output (b) Menu in the software (c) Digit '2' Recognized by CNN

(d) Digits '73' Recognized by CNN (e) Digits '018' Recognized by CNN

Fig. 4.1 Images of software

5. RESULTS

During the experimentation execution of sigmoid, softmax, relu and tanh actuation capacities are tested. Results are taken with various enhancers specifically Adam, Adagrad, Adadelta, SGD and RMSprop. Table 5.1 shows the Performance after using different optimizers.

Table 5.1 Performance of Different optimizers

		MNIST dataset	
Optimizer	epoch	Training accuracy	Testing accuracy
SGD	50	99.64%	99.34%
RMSprop	50	99.21%	99.11%
Adagrad	50	99.97%	98.18%
	60	99.62%	98.09%
Adadelta	50	99.86%	99.27%
	70	99.47%	99.15%
	100	99.44%	99.20%
Adam	50	99.02%	98.05%
	60	98.88%	98.74%
	70	99.73%	98.01%
	80	99.07%	99.43%
	90	98.56%	99.19%

Table 5.2 shows the outcome with changing batch size. Sigmoid activation function, 0.1 dropout and Adam optimizer is utilized. Expanding batch size is not having effect on testing/training precision.

Table 5.2 Batch size experimentation

	MNIST dataset	
Batch Size	Training accuracy	Testing accuracy
16	98.40%	98.78%
32	97.13%	97.98%
100	98.36%	97.86%
120	98.11%	97.73%
140	98.83%	98.49%
150	97.54%	98.01%
160	98.73%	97.29%
200	97.83%	98.79%

Table 5.3 shows impact of adam optimizer, sigmoid activation function, 100 Batch size, 60 epochs with fluctuating learning rate. Accuracy on MNIST dataset is greater than 98%-99% for most of cases.

Table 5.3 Learning rate experimentation

	MNIST dataset	
Learning Rate	Training accuracy	Testing accuracy
0.01	98.99%	98.84%
0.02	96.97%	96.95%
0.001	99.50%	99.29%
0.002	98.06%	96.41%

6. ACKNOWLEDGMENT

We would wish to thank our Principal Dr. R. P. Singh, Academic Dean Dr. R. B. Barjibhe and Head of Department, Dr. D. D. Patil for opening the doors of knowledge towards the realization of in this work, we also wish to express true sense of gratitude towards Prof. A. P. Ingale who at every discrete step in study of this work and contributed his valuable guidance and help us to solve every problem that arose. Most likely We would like to express my sincere gratitude towards my family and friends for always being there when we needed them the most. With all respect and gratitude, we would like to thank all authors listed and not listed in references whose concepts are studied and used us whenever required. We owe our all success to them.

REFERENCES

[1] A. D. Anuj Dutt, "Handwritten digit recognition using deep learning," IJARCET, vol. 6, pp. 990–997, July 2017.

[2] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard, "Handwritten digit recognition: applications of neural network chips and automatic learning," IEEE Communications Magazine, vol. 27, no. 11, pp. 41–46, 1989.

[3] R. Kumar, M. K. Goyal, P. Ahmed, and A. Kumar, "Unconstrained handwritten numeral recognition using majority voting classifier," in 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012, pp. 284–289.

[4] P. AHMED and C. Y. SUEN, "Computer recognition of totally unconstrained handwritten zip codes," International Journal of Pattern Recognition and Artificial Intelligence, vol. 01, no. 01, pp. 1–15, 1987. [Online]. Available: https://doi.org/10.1142/S0218001487000023

[5] S. Knerr, L. Personnaz, and G. Dreyfus, "Handwritten digit recognition by neural networks with single-layer training," Neural Networks, IEEE Transactions on, vol. 3, pp. 962 – 968, 12 1992.

[6] "Mnist handwritten digit database, yann lecun, corinna cortes and chris burges." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[7] C. to Wikimedia projects, "Mnist database,"May 2021. [Online]. Available: https://en.wikipedia.org/wiki/MNIST database

[8] B. Zhang and S. Srihari, "Fast k-nearest neighbor classification using cluster-based trees," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 4, pp. 525–528, 2004.

[9] E. Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on mnist database," Image and Vision Computing, vol. 22, no. 12, pp. 971–981, 2004, proceedings from the 15th International Conference on Vision Interface. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885604000721

[10] F. Siddique, S. Sakib, and M. A. B. Siddique, "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers," in 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019, pp. 541–546.

[11] "An introduction to convolutional neural networks." [Online]. Available: https://www.researchgate.net/publication/285164623 An Introduction to Convolutional Neural Networks

[12] A. G. Chigozie Enyinna Nwankpa, Winifred Ijomah and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning." [Online]. Available: https://arxiv.org/pdf/1811.03378.pdfs

[13] R. Dixit, R. Kushwah, and S. Pashine, "Handwritten digit recognition using machine and deep learning algorithms," 07 2020.

[14] D. Dataman, Towards Data Science, Jun 2021. [Online]. Available: https://towardsdatascience.com/convolutional-autoencoders-for-image-noise-reduction-32fce9fc1763

[15] C. to Wikimedia projects, "Noise reduction," Jun 2021. [Online]. Available: https://en.wikipedia.org/wiki/Noise reduction.