International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 3 Mar 2022, pp: 928- 934 www.ijaem.net ISSN: 2395-5252

Survey on Handwritten Digit Recognition using Machine Learning

Uma Kavya, Dr.Ganesh D
Department of Master's Computer Application,Jain(Deemed-to-be University),Bengaluru-560069

Submitted: 10-03-2022 Revised: 21-03-2022 Accepted: 25-03-2022

ABSTRACT
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 reduce in recognizing, learning, predictions and many more areas. This paper presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset, comparing classifiers like KNN, PSVM, NN and convolution neural network on basis of performance, accuracy, time, sensitivity, positive productivity, and specificity with using different parameters with the classifiers.

Keywords— Handwritten Digits, Vector Machine, Neural Networks, Convolution, Machine Learning

I.INTRODUCTION

For a beginner aspirant, starting hurdle in the field of deep learning and machine learning is the MNIST dataset for Handwritten Digit Recognition and this system involves understanding and recognition of 10 handwritten digits (0- 9) by a machine. Handwritten Digit Recognition from the MNIST dataset has been very popular among researchers as by using various classifiers for different algorithms and parameters, the error rate has been reduced a lot such as from linear classifier (1-layer NN) with 12% to 0.23% by a committee of 35 convolution neural networks (YannLeCun, the MNIST database of Handwritten Digits). The scope of this study is to build an offline Handwritten Digit Recognition system and compare the different classifiers and combination methods by focusing on to achieve near the human performance. The handwritten digits are not always of the same size, thickness, or orientation and position relative to the margins. For a task of writing different digits (0-9) for different persons the general problem faced would be of digit classification problem and the similarity between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 etc. Also people write the same digit in many different ways. Finally, the uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the digits [5]. Here comes the use of deep learning and machine learning. In recent years, deep learning and machine learning have become necessary for image processing, object detection, handwritten digit recognition, character recognition, segmenting images, and building automated machines which might process on their own[11][12][13][14][15].

II. MNIST DATASET

A. Understanding Dataset and Format.

Samples provide by MNIST (Modified National Institute of Standards and Technology) dataset consists of handwritten digits with a training set of 60,000 examples and a test set of 10,000 labeled images. This is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. The images in the MNIST dataset are provided in form of 28x28 gray scale intensities of images representing an image along with labels. This is also the same in case of the 10,000 testing images (YannLeCun, the MNIST database of Handwritten Digits). The MNIST problem is a dataset developed by YannLeCun, Corinna Cortes and Christopher Burges for evaluating machine learning models on the handwritten digit classification problem (YannLeCun, the MNIST database of Handwritten Digits). The dataset was made from a number of scanned document dataset available from the National Institute of Standards and Technology (NIST). This is where the name was given as the Modified NIST or MNIST dataset. Images of digits were taken from various scanned digits, normalized in size and justify as centered. This makes it an excellent dataset for evaluating models and allowing the machine learning aspirant to focus on deep learning and machine learning with very little data cleaning (Machine Learning Mastery, CNN with Keras.). It is a digit recognition problem, as such there are 10 digits (0 to 9) or simply 10 classes to predict from. The first 5,000 examples of the test set are taken from the original NIST training set and remaining from the originalNIST test set. The first 5,000 are cleaner and easier than remaining (YannLeCun, the MNIST database of Handwritten Digits). There are 4 files in the dataset:

•train-images-idx3-ubyte: training set images
•train-labels-idx1-ubyte: training set labels
•t10k-images-idx3-ubyte: test set images
•t10k-labels-idx1-ubyte: test set labels Note –

• Pixels are arranged row-wise, ranging from 0 to 255, as from RGB color code.
• Background as white (0 value from RGB) and foreground as black (255 value from RGB).
• Labels of digits classified from 0 to 9. Above points are also understandable by changing the format to csv, as more neat, clean and easily understandable format gives these details.

B. Understanding Dataset

So, before starting further research, the better point should be to get familiar with the provided dataset. Here both the training and testing images and labels have the first two columns consisting of the 'magic number' and the number of items in the file (YannLeCun, the MNIST database of Handwritten Digits). The magic number has its first two bytes equal to zero. This magic number is read as MSB first and its format is as shown [1]:

TRAINING SET IMAGE FILE (train-images-idx3-ubyte):

[offset]	[type]	[value]	[description]
0000	32 bit integer	0x00000803(2051)	magic number
0004	32 bit integer	60000	number of images
0008	32 bit integer	28	number of rows
0012	32 bit integer	28	number of columns
0016	unsigned byte	??	pixel
0017	unsigned byte	??	pixel
....			
xxxx	unsigned byte	??	pixel

Figure 1 Training set image file format

TRAINING SET LABEL FILE (train-labels-idx1-ubyte):

[offset]	[type]	[value]	[description]
0000	32 bit integer	0x00000801(2049)	magic number (MSB first)
0004	32 bit integer	60000	number of items
0008	unsigned byte	??	label
0009	unsigned byte	??	label
....			
xxxx	unsigned byte	??	label

Figure 2 Training set label file format

Table 1–Magic number format

2 Bytes	1 Byte	1 Byte
00	Data Type	Dimensions

From figure 1, the magic number value for that images is 0x00000803(2051) and from figure 2, magic number is 0x00000801 (2049) value for labels.

Table 2 - Magic number for specified data - 0x00000803(2051) and 0x00000801 (2049)

2 Bytes	1 Byte	1 Byte
00 00	08	03
00 00	08	01

This gives us the following information:
1. The 0000 (2 bytes) informing the beginning of the file.
2. 08 tells us that third byte is of unsigned byte type.
3. The fourth byte, 03 tells us that the matrix has three dimensions and 01 informing with just one dimension.

The third byte represents whether the data is an integer, float, short, long or unsigned type. The fourth byte tells the dimension of the vector or matrix i.e. the number of rows and columns. If it is equal to 1, then it's a vector else it is a matrix. The number of items variable is also read as MSB first. [1] B. IDX Format As our dataset is available in IDX format (YannLeCun, the MNIST database of Handwritten Digits), we can change our dataset into csv formats by algorithm (Joseph Chet Redmon, Algorithm to change idx into csv])and we can achieve MNIST dataset in csv format. The format of these is easy to understand:

- The first value is the "label", that is, the actual digit that the handwriting is supposed to represent, such as a "7" or "9". It is the answer to which the classifier is aspiring to classify.
- The subsequent values, all comma separated, are the pixel values of the handwritten digit. The size of the pixel array is 28 by 28, so there are 784 values after the label. (Joseph Chet Redmon, Algorithm to change idx into csv).

III. SURVEY OF HANDWRITTEN DIGIT RECOGNITION

Table 3: Review of Handwritten Digit Recognition

<table>
<tr><th>S.No.</th><th>Paper Name</th><th>Review</th></tr>
<tr><td>[1]</td><td>Handwritten Digit Recognition Using Deep Learning</td><td>Accuracy and time comparison between machine learning (RFC, KNN, SVM) and deep learning (Multilayer CNN) on MNIST dataset. Below calculations on CPU, for more accuracy, reduced training and testing time GPU might be useful and GPU can help in getting parallelism and attaining much better results.
<table>
<tr><td>Accuracy Comparison Table</td><td>RFC</td><td>KNN</td><td>SVM</td><td>CNN</td></tr>
<tr><td>Trained classifier accuracy (in %)</td><td>99.71</td><td>99.71</td><td>99.71</td><td>99.71</td></tr>
<tr><td>Accuracy on Test Images (in %)</td><td>96.89</td><td>96.67</td><td>97.91</td><td>98.72</td></tr>
<tr><td>Training Time (in min)</td><td>10</td><td>15</td><td>14</td><td>70</td></tr>
<tr><td>Testing Time (in min)</td><td>6</td><td>9</td><td>10</td><td>20</td></tr>
</table></td></tr>
<tr><td>[2]</td><td>Handwritten Digit Recognition using Proximal Support Vector Machine</td><td>A system based on Histogram of Oriented Gradient (HOG) feature. Used proximal support vector machine over standard SVM classifier as this takes less time and performance of PSVM classifier is better than artificial neural network.
Total of 20,000 samples taken for both training and testing data (1,000 samples for a digit)
10 class linear PSVM got 98.65% with training 59 milliseconds.
The system has also maintained small dimension for feature vector without including an additional dimensionality reduction and less training time.</td></tr>
<tr><td>[3]</td><td>Intelligent Handwritten Digit Recognition using Artificial Neural Network</td><td>A dataset of 5,000 examples from MNIST, was trained with an algorithm of gradient descent back propagation and then tested with an algorithm of feed-forward with the number of hidden units and iterations and accuracy achieved was 99.32%.
Multilayer Perception (MLP) neural network with 35 neurons and 250 iterations were found. The proposed system gave 99.32% accuracy on training and 100% training accuracy.</td></tr>
<tr><td>[4]</td><td>Recognition of Handwritten Digits using Proximal Support Vector Machine</td><td>The accuracy of 98.65% with PSVM and reduced time from 109 seconds (by ANN) to 59 milliseconds for PSVM classifier on 10,000 samples for each of training and training set respectively.
<table>
<tr><td>Parameters</td><td>ANN (100 epoch)</td><td>PSVM</td></tr>
<tr><td>Sensitivity (%)</td><td>91.84</td><td>93.22</td></tr>
<tr><td>Positive predictivity (%)</td><td>91.87</td><td>93.27</td></tr>
<tr><td>Specificity (%)</td><td>99.09</td><td>99.25</td></tr>
<tr><td>Accuracy (%)</td><td>98.37</td><td>98.65</td></tr>
</table></td></tr>
</table>

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 3 Mar 2022, pp: 928- 934 www.ijaem.net ISSN: 2395-5252

		Training time (classifier) in seconds	109	0.059
[5]	Handwritten Digits Recognition	This report presents the implementation of Principal Component Analysis (PCA) combined with a 1-nearest neighbor and achieving 78.4% accuracy on MNIST dataset. Reason for inaccuracy was the most basic algorithm for feature selection and classification. Therefore, it is very hard to get good results compared to a more complex system. To improve accuracy and better results, examples in both the training and testing should be increased. There are such samples where it is even difficult for a human to classify, similarities between digits is much higher in some images.		
[6]	A trainable feature extractor for handwritten digit recognition	Accuracy comparison between NN and SVM classifiers with various kernels.		
		Classifier	Distortion	Test error (%)
		TFE – SVM		0.83
		LeNet5	Elastic	0.72
		LeNet5	Affine	0.68
		TFE – SVM	Elastic	0.56
		TFE – SVM	Affine	0.54
[7]	Multi-column Deep Neural Networks for Image Classification	With the use of classifier of the committee of 35 conv. net, 1-20-P-40-P-150-10 [elastic distortions] and pre-processing of width normalization 0.23% accuracy can be achieved from these multi-column deep neural networks (MCDNN).		
[8]	Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition	Neural network has achieved an error rate, simply defined as 35 out of 10,000 digits misclassified or 0.35% error percentage, with 6-layer NN architecture or number of neurons in each layer (784-2500-2000-1500-1000-500-10) (on GPU) [elastic distortions] and no pre-processing involved.		
[9]	Best Practices for Convolution neural networks Applied to Visual Document Analysis	Split training set in 50,000 for training, 10,000 for validation and parameter adjustments and results on test set with optimal parameters values on validation.		
		Algorithm	Distortion	Error (%)
		2 Layer MLP (CE)	None	1.6
		2 Layer MLP (CE)	Elastic	0.7
		2 Layer MLP (CE)	Affine	1.1
		Simple Conv. (CE)	Elastic	0.4
[10]	Gradient-Based Learning Applied to Document Recognition	Most of the early benchmarks achieved by using pre-processing on algorithms.		
		Algorithm	Pre-processing	Error (%)

		K-NN, tangent distance	subsampling to 16x16 pixels	1.1
		K-nearest-neighbors, Euclidean (L2)	None	5.0
		SVM deg 4 polynomial	Deskewing	1.1
		Virtual SVM deg-9 poly [distortions]	None	0.8
		3-layer NN, 500+150 hidden units	None	2.95
		Convolutional net LeNet-5, [no distortions]	None	0.95
		Convolutional net LeNet-5, [huge distortions]	None	0.85

IV. CONCLUSION

The performance of the classifier can be measured in terms of ability to identify a condition properly (sensitivity), the proportion of true results (accuracy), number of positive results from the procedure of classification as false positives (positive predictions) and ability to exclude condition correctly (specificity) [4]. In this survey paper, we looked at different classifiers used with different features and parametric values performing with various accuracies and error rate. Classifiers ask nearest neighbors (KNN), proximal support vector machine, and neural network with different layers can perform well, but best performing classifier on MNIST dataset is convolution neural network (part of deep learning) and the performance is best in this classifier [10].

REFERENCES

[1]. AnujDutt, AashiDutt, 'Handwritten Digit Recognition Using Deep Learning', (IJARCET) Volume 6, Issue 7, July 2017, ISSN: 2278 – 1323, Page number – 990-997

[2]. Priya, Rajendra Singh, Dr. SoniChanglani, 'Handwritten Digit Recognition using Proximal Support Vector Machine' , Journal of Emerging Technologies and Innovative Research, Volume 4, Issue 04,April 2017, ISSN Number: 2349-5162, Page No: 251-254

[3]. Saeed AL-Mansoori, 'Intelligent Handwritten Digit Recognition using Artificial Neural Network', Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 5, Issue 5, (Part -3) May 2015, pp.46-51

[4]. SwapnaPravaEkka, 'Recognition of Handwritten Digits using Proximal Support Vector Machine', Department of ECE, EThesis NIT, Rourkela, 2014, ID Code - 6066

[5]. Gaurav Jain, Jason Ko, 'Handwritten DigitsRecognition', Project Report, University of Toronto, 11/21/2008

[6]. [6] Fabien Lauera,,ChingY. Suenb, Gérard Blocha, 'A trainable feature extractor for handwritten digit recognition', Pattern Recognition Society. Published by Elsevier Ltd., Volume 40, Issue 6, June 2007, Pages 1816-1824, October 2006.

[7]. Dan Cires¸anUeli Meier, JurgenSchmidhuber, 'Multi-column Deep Neural Networks for Image Classification', Journal reference: CVPR 2012, p. 3642-3649 , Report number: IDSIA-04-12 Feb 2012

[8]. Dan ClaudiuCiresan, Ueli Meier, Luca Maria Gambardella, JurgenSchmidhuber, 'Deep Big Simple Neural Nets Excel on HandwrittenDigit Recognition', Neural Computation, Volume 22, Number 12, December 2010

[9]. Patrice Y. Simard, Dave Steinkraus, John C. Platt, 'Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis', Institute of Electrical and Electronics Engineers, Inc.,Inproceedings, ISBN: 0-7695-1960-1, Sept., 2003

[10]. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, 'Gradient-Based Learning Applied to Document Recognition', Proceedings of the IEEE, v. 86, pp. 2278-2324, 1998.

[11]. JurgenSchmidhuber, 'Deep Learning in Neural Networks: An Overview', Technical Report IDSIA-03-14, April 2014.

[12]. Berend-Jan van der Zwaag, 'Handwritten Digit Recognition:A Neural Network

Demo', B. Reusch (Ed.): Fuzzy Days 2001, LNCS 2206, pp. 762–771, 2001.
[13]. Haider A. Alwzwazy, Hayder M. Albehadili, Younes S. Alwan, Naz E. Islam, 'Handwritten Digit Recognition Using Convolutional Neural Networks', IJIRCCE, Vol. 4, Issue 2, February 2016.
[14]. Stefan Knerr, LConPersonnaz, and GCrard Dreyfus, 'Handwritten digit recognition by NeuralNetworks with Single-Layer Training', IEEE Transactions On Neural Networks, VOL. 3, NO. 6, Nov., 1992
[15]. Faisal Tehseen Shah, Kamran Yousaf, 'Handwritten Digit Recognition Using Image Processing and Neural Networks', WCE 2007, VOL-1, July 2 - 4, 2007