A Comparative Study on Handwriting Digit Recognition Using Neural Networks

Mahmoud M. Abu Ghosh
Faculty of Information Technology
Islamic University of Gaza
Palestine
ma.abughosh@students.iugaza.edu.ps

Ashraf Y. Maghari
Faculty of Information Technology
Islamic University of Gaza
Palestine
amaghari@iugaza.edu.ps

***Abstract*—The handwritten digit recognition problem becomes one of the most famous problems in machine learning and computer vision applications. Many machine learning techniques have been employed to solve the handwritten digit recognition problem. This paper focuses on Neural Network (NN) approaches. The most three famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN). In this paper, the three NN approaches are compared and evaluated in terms of many factors such as accuracy and performance. Recognition accuracy rate and performance, however, is not the only criterion in the evaluation process, but there are interesting criteria such as execution time. Random and standard dataset of handwritten digit have been used for conducting the experiments. The results show that among the three NN approaches, DNN is the most accurate algorithm; it has 98.08% accuracy rate. However, the execution time of DNN is comparable with the other two algorithms. On the other hand, each algorithm has an error rate of 1–2% because of the similarity in digit shapes, specially, with the digits (1,7) , (3,5) , (3,8) , (8,5) and (6,9).**

Keywords —Handwriting Digit Recognition; Neural Network; CNN; DNN; DBN

I. Introduction

Nowadays, more and more people use images to represent and transmit information. It is also popular to extract important information from images. Image recognition is an important research area for its widely applications[1, 2]. In the relatively young field of computer pattern recognition, one of the challenging tasks is the accurate automated recognition of human handwriting. Indeed, this is precisely a challenging problem because there is a considerable variety in handwriting from person to person. Although, this variance does not cause any problems to humans, yet, however it is more difficult to teach computers to recognize general handwriting [3]. For the image recognition problem such as handwritten classification, it is very important to make out how data are represented in images[1]. The data here is not the row pixels, but should be the features of images which has high level representation[2, 4]. For the problem of handwritten digit recognition, the digit's structure features should be first extracted from the strokes. Then the extracted features can be used to recognize the handwritten digit. The high performance of large-scale data processing ability is the core technology in the era of big data. Most current classification and regression machine learning methods are shallow learning algorithms [4]. It is difficult to represent complex function effectively, and its generalization ability is limited for complex classification problems[5, 6]. Deep learning is a multilayer neural network learning algorithm which emerged in recent years. Applications of deep learning to various problems have been the subject of a number of recent studies ranging from image classification and speech recognition to audio classification [5, 7-9]. It has brought a new wave to machine learning, and making artificial intelligence and human-computer interaction advance with big strides. Deep Learning algorithms are highly efficient in image recognition tasks such as MNIST digit recognition[10].

In this paper, we apply deep learning algorithms to handwritten digit recognition, and explore the three mainstream algorithms of deep learning; the Convolutional Neural Network (CNN), the Deep Belief Network (DBN) and the Deep Neural Network (DNN)[4].

II. BACKGROUND

In this section, we give an overview of the three algorithms and the tools employed in our paper: -

A. Convolutional Neural Network (CNN):

A simple CNN model can be seen in Fig. 1. The first layer is the input layer; the size of the input image is 28 × 28. The second layer is the convolution layer C2, it can obtain four different feature maps by convolution with the input image. The third layer is the pooling layer P3. It computes the local average or maximum of the input feature maps [11].

The next convolution layer and pooling layer operate in the same way, except the number and size of convolution kernels. The output layer is full connection; the maximum value of output neurons is the result of the classifier in end [12].

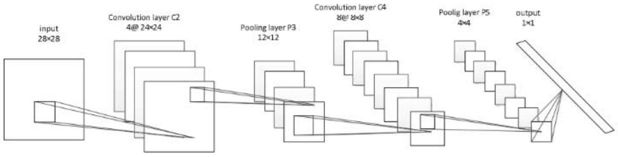

Fig. 1. A simple structure of CNN [13] .

978-1-5386-2269-8/17 $31.00 © 2017 IEEE
DOI 10.1109/ICPET.2017.20

B. *Deep Belief Network (DBN):*

Deep Belief Network is a probability generation model, and belongs to unsupervised learning algorithms [12]. It consists of multiple Restricted Boltzmann Machine (RBM). RBM is an effective feature extraction method that gives DBN the ability to extract more abstract features by stacking multiple RBM [14]. A typical DBN structure is shown in Fig. 2.

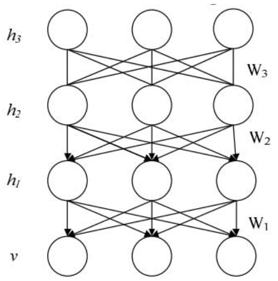

Fig. 2. The structure of DBN [13]

C. *Deep Neural Network (DNN):*

The initially random weights of DNN are iteratively trained to minimize the classification error on a set of labeled training images; generalization performance is then tested on a separate set of test images[4]. DNN has 2-dimensional layers of winner-take-all neurons with overlapping receptive fields whose weights are shared. Given some input pattern, a simple max pooling technique determines winning neurons by partitioning layers into quadratic regions of local inhibition, and selecting the most active neuron of each region[8, 15]. The winners of some layer represent a smaller, down-sampled layer with lower resolution, feeding the next layer in the hierarchy [15]. The approach is inspired by Hubel and Wiesel's seminal work on the cat's primary visual cortex, which identified orientation selective simple cells with overlapping local receptive fields and complex cells performing down-sampling-like operations is shown in Fig. 3 [15].

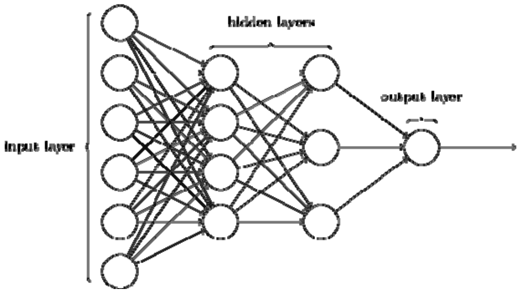

Fig. 3. The structure of DNN[7] .

D. *Neural Network Toolbox in matlab (simulate):*

Neural Network Toolbox provides algorithms, functions, and apps to create, train, visualize, and simulate neural networks[16, 17]. The toolbox includes convolutional neural network and auto encoder deep learning algorithms for image classification and feature learning tasks by used MATLAB programming language [17, 18].

III. LITERATURE REVIEW & RELATED WORK

Wu et al. have applied deep learning to the real-word handwritten character recognition, and obtained good performance for image recognition. They analyzed the different between CNN and DBN by comparing the experiment results. Deep learning can approximate the complex function through deep nonlinear network model. It does not only avoid the large workload of manually extract features, but also it is better to describe potential information of the data [13]. However, they did not consider the evaluation factors as execution time.

"Kaensar et al. have concluded that different classifier affects the recognition rate for handwritten digit recognition. Accordingly, they applied three classification techniques by using the open source Weka tool kit for training and testing the dataset which was obtained from the UCI repository. The presented results show that SVM is the best classifier to recognize handwritten digits. However, the main problem of the SVM classifier is the time consuming of the training process. Conversely, other methods like neural networks give insignificantly worse results, but their training is much quicker [19]."

Saabni et al. have presented an algorithm that trains k-sparse auto encoders and used their hidden layers to be stacked as retrained hidden layers into a deep neural network. Their proposed system is a part of a more complex system which aims to analyze images of checks in order to extract and recognize important information such as amounts and texts from checks images. To avoid training of deep layer with the back propagation algorithm directly on randomized weights, the first two layers have been trained a side using sparse auto encoders to extract important features in hierarchical manner [20].

Our work, however, is different from the reported related works in the context that we compare three algorithms depending on four factors including accuracy, performance and execution time. While, to the best of our knowledge, most of the related works focused on the accuracy.

IV. METHODOLOGY

As shown in Fig. 4, our proposed system could be divided into five main steps: preprocessing, segmentation, feature extraction, classification, training and recognition as shown in Fig. 4. The stages are:

A. Preprocessing

At this stage, because all images in the database are clean and without noise, no noise reduction technique is required here. But in a real system we need to remove noise from the images. In any document there could be optical noises present along with the documents. Especially in the handwritten documents the character shapes may not be always unique. Hence the preprocessing is mandatory. We will first apply an Erosion with 3 X 3 structuring elements which will eliminate the one bit errors and give a smooth edge. Then the characters are dilated with 2 X 2 elements.

B. Segmentation

After the preprocessing step, an image of sequence of digit is decomposed into sub-images of individual digit. Preprocessed input image is segmented into isolated digit by assigning a number to each digit using a labeling process. This labeling provides information about number of digits in the image. Each individual digit is uniformly resized into 100 X 70 pixels for classification and recognition stage [16].

C. Feature Extraction

After the segmentation step, the Segmented Image is given as input to feature extraction module. The statistical features of the histogram; mean and standard deviation, will be extracted from the images.

D. Training

After the Feature Extraction step, each of the proposed algorithms (CNN, DBN, DNN) is trained separately with the training images.

E. Classification & Recognition

After the training step, “the classification & Recognition stage is the decision making part of a recognition system and it uses the features extracted in the previous stage. A feed forward back propagation neural network having two hidden layers with architecture of 54-100-100-38 is used to perform the classification. The hidden layers use log sigmoid activation function, and the output layer is a competitive layer, as one of the digits is to be identified. The feature vector is denoted as X where X = (f1, f2,…,fd) where f denotes features and d is the number of zones into which each digit is divided. The number of input neurons is determined by length of the feature vector d. The total numbers of digits’ n determine the number of neurons in the output layer. The number of neurons in the hidden layers is obtained by trial and error [16]. The most Compact network is chosen and presented as shown in Fig. 4.
It is to recognize handwritten digits using the three algorithms in which each algorithm recognizes the image in its own way process. After the training process, the Digits are compared by an expert to assess the accuracy of the tip. Also, the precision, the expense of performance and execution time are compared.

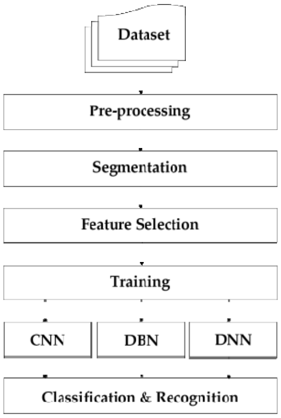

Fig. 4. A block diagram of proposed model.

V. EXPERIMENTAL

A. Dataset

1) Standard Dataset

The primary dataset that is used in training the classifier is the MNIST dataset published by Yann LeCun of Courant Institute at New York University. The dataset contains a 60,000 labeled training set and a 10,000 labeled test set[21].

The handwritten data samples come from approximately 250 different writers and completely different writers were sampled for the test set. That is, there is no intersection between the writers of the test set and training set [10, 21, 22].

The MNIST dataset is available as binary files stored in an IDX file format and a visualization of what the numbers look like can be seen in Fig. 5. A much bigger visualization of the data set is also available in the appendix. The preprocessing done was to read in the data into an image matrix and perform a basic normalization procedure for each image where by each pixel value was divided by the maximum pixel value for that sample image [10, 21, 22].

Fig. 5. A sample of the MNIST [10, 21, 22].

2) Random Dataset

This random dataset contains 85 different digit made by the author and collected from different resource. Fig. 6 shows a sample of the random dataset.

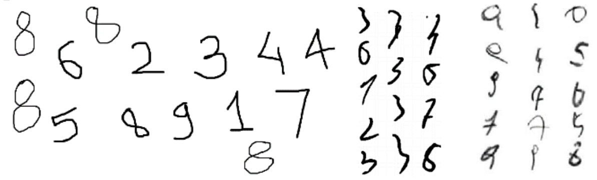

Fig. 6. A sample of the random dataset.

B. Experimental Environment

- Windows 10 operating system as test platform,
- CPU is Intel Core I7 6500u, which has dual cores running on 2.4GHZ.
- RAM 16GB.
- GPU is Nvidia GTX 970, CUDA Cores 1664, 4GB GDDR5.

C. *Experiment's Evaluation Factors*

The three algorithms are evaluated in accordance with the proposed model according to the following Factors:

- Accuracy
- Performance
- Execution Time

VI. EXPERIMENTAL RESULT & DISCUSSION

In this section, we will show and discuss the results of the experiment, for Standard and Random Dataset:

A. *Standard Dataset*

The results show here superiority of CNN to recognize the digit (0) accurately as shown in Fig. 7. The recognition accuracies for the other algorithms are also high due to the ease of writing digit (0). In addition, CNN is superior in the performance as shown in Fig. 8 and Fig. 9. However, the constant digit (0) has the form of a circular shape in all cases as shown in Fig. 5.

For recognizing digit (1), the results show superiority of CNN in accuracy as shown in Fig. 7. In addition, CNN is superior in the performance as shown in Fig. 8, but DBN has superiority in the time of execution as shown in Fig. 9. However, a problem of similarity may occur with the digit (7) as shown in Fig. 5.

For digit (3), the results show superiority for DBN in term of accuracy as shown in Fig. 7. In addition, DNN is superior in the performance as shown in Fig. 8, and at the time of execution as shown in Fig. 9. But a problem of similarity occurs sometimes with the digit (5), (8) as shown in Fig.5.

Regarding to digit (6), DNN is the most accurate algorithm in recognizing digit (6) as shown in Fig. 7. The other recognition accuracies for the other algorithms are also high due to the ease of writing digit (6). DNN is also superior in the performance as shown in Fig. 8, and at the time of execution as shown in Fig. 9. But there is a problem that sometimes occurs because of similarity with the digit (9) as shown in Fig. 5.

In all the algorithms, the accuracy of recognition depends on the ratio of the similarity in shape and sometimes depends on the working principle of the algorithm.

Fig. 8 and Fig. 9 show the evaluation results of handwritten digit recognition in terms of Performance and Execution Time, respectively. The figures show that DNN outperforms the other algorithms for all factors; Performance and Execution Time. This result is also consistent with previous studies. It is shown that is it possible to build a digit classification with a sufficiently high accuracy using only basic machine learning techniques [3, 15, 19, 20, 23].

A. *Accuracy: -*

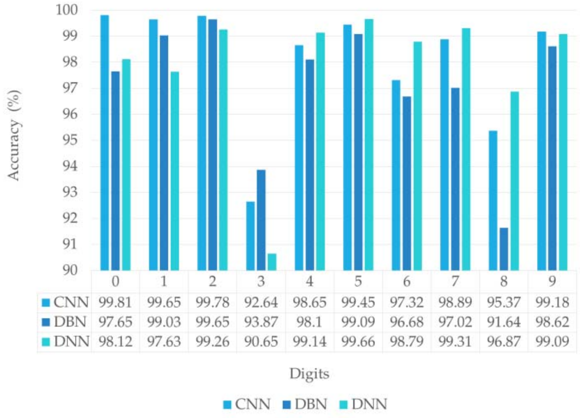

Fig. 7. Accuracy of Different Digits by (CNN, DBN, DNN)

B. *Performance: -*

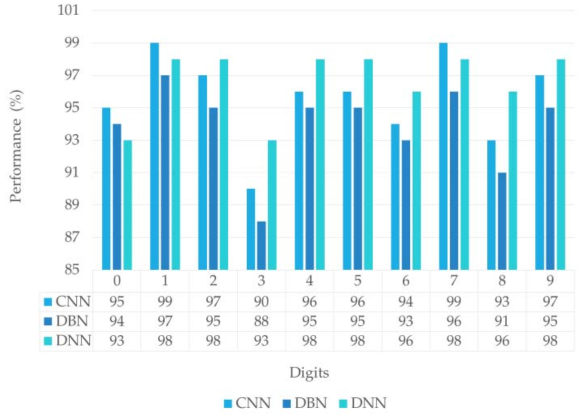

Fig. 8. Performance of Different Digits by (CNN, DBN, DNN)

C. *Execution Time: -*

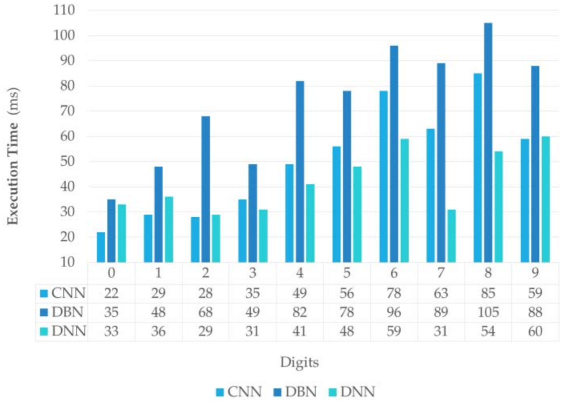

Fig. 9. Execution Time of Different Digits by (CNN, DBN, DNN)

B. Random Dataset

To prove the efficiency and validity of the results obtained by standard dataset, a random dataset is used.

Table 1 shows the recognition results of some experiments conducted on Random Dataset. The results show significant convergence in the accuracy of DNN algorithm compared to the previous results shown Fig. 7. This also affects the performance ratio and execution time because the confusion happens in recognition processing on the similarity between the two digits, such as (3) and (8). This is due to lack of trained algorithms on previously forms of the digits as (7), (3) and (5).

TABLE I. SOME OF EXPERIMENT RESULTS FOR RANDOM DATASET.

Digit	Algo.	Accuracy	Performance	Execution Time
1	CNN	98. 45%	95. 11%	37ms
	DBN	97. 01%	93. 38%	45ms
	DNN	98.6 0%	97. 13%	31ms
7	CNN	97. 68%	97. 61%	36ms
	DBN	97. 04%	97. 04%	51ms
	DNN	98. 78%	97. 78%	34ms
3	CNN	93. 98%	91. 32%	71ms
	DBN	90. 61%	90. 01%	77ms
	DNN	95. 43%	92.65 %	65ms
5	CNN	95. 99%	95. 98%	44ms
	DBN	95. 78%	93. 04%	77ms
	DNN	96. 41%	95. 63%	49ms
8	CNN	96. 36%	96. 35%	79ms
	DBN	96. 10%	94. 43%	73ms
	DNN	97. 33%	96. 47%	68ms
6	CNN	98. 41%	97. 91%	57ms
	DBN	98. 79%	97. 87%	51ms
	DNN	99. 48%	98.10%	53ms
9	CNN	97. 93%	97.65 %	67ms
	DBN	97. 42%	96. 32%	62ms
	DNN	98. 29%	97. 92%	59ms

VII. CONCLUSION

In this paper, we compared three Neural Network based recognition algorithms to determine the best algorithm in terms of many factors such as accuracy and performance. Other criteria such as execution time have been also taken in consideration. Random and standard datasets of handwritten digit have been used to evaluate the algorithms. The results showed that DNN is the best algorithm in terms of accuracy and performance. CNN algorithm and DNN are of almost equal in terms of accuracy. DNN algorithm, however, was better than CNN and DBN in terms of execution time. By recognizing the correct digits, the margin of errors may occur with similarities between the digits.

VIII. FUTURE WORK

Future efforts can study the optimization of deep learning, and apply it to more complex image recognition problems. It is interesting is to look at building a real-time classifier and a related application (mobile and/or desktop) that will take in user input and immediately do recognition and convert that to a digit (1,7), (3,5) , (3,8) , (8,5) and (6,9).

REFERENCES

[1] Ting, R., S. Chun-lin, and D. Jian, Handwritten character recognition using principal component analysis. MINI-MICRO Systems, 2005. 26(2): p. 289-292.

[2] Walid, R. and A. Lasfar. Handwritten digit recognition using sparse deep architectures. in Intelligent Systems: Theories and Applications (SITA-14), 2014 9th International Conference on. 2014. IEEE.

[3] Li, Z., et al. Handwritten digit recognition via active belief decision trees. in Control Conference (CCC), 2016 35th Chinese. 2016. IEEE.

[4] Schmidhuber, J., Deep learning in neural networks: An overview. Neural Networks, 2015. 61: p. 85-117.

[5] LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. Nature, 2015. 521(7553): p. 436-444.

[6] Hinton, G.E. and R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks. Science, 2006. 313(5786): p. 504-507.

[7] Yu, K., et al., Deep learning: yesterday, today, and tomorrow. Journal of computer Research and Development, 2013. 50(9): p. 1799-1804.

[8] Sun, Z.-J., et al., Overview of deep learning. Jisuanji Yingyong Yanjiu, 2012. 29(8): p. 2806-2810.

[9] Bengio, Y., Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2009. 2(1): p. 1-127.

[10] LeCun, Y., C. Cortes, and C.J. Burges, The MNIST database of handwritten digits. 1998.

[11] Bouchain, D., Character recognition using convolutional neural networks. Institute for Neural Information Processing, 2006. 2007.

[12] Hinton, G.E., S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets. Neural computation, 2006. 18(7): p. 1527-1554.

[13] Wu, M. and L. Chen. Image recognition based on deep learning. in Chinese Automation Congress (CAC), 2015. 2015. IEEE.

[14] Fischer, A. and C. Igel. An introduction to restricted Boltzmann machines. in Iberoamerican Congress on Pattern Recognition. 2012. Springer.

[15] Ciregan, D., U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. 2012. IEEE.

[16] Lapuschkin, S., et al., The LRP toolbox for artificial neural networks. Journal of Machine Learning Research, 2016. 17(114): p. 1-5.

[17] Neural Network Toolbox - MATLAB & Simulink. Available from: https://www.mathworks.com/products/neural-network.html.

[18] Beale, M.H., M.T. Hagan, and H.B. Demuth. Neural network toolbox™ user's guide. in R2012a, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098,, www. mathworks. com. 2012. Citeseer.

[19] Kaensar, C. A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor. in The 9th International Conference on Computing and InformationTechnology (IC2IT2013). 2013. Springer.

[20] Saabni, R. Recognizing handwritten single digits and digit strings using deep architecture of neural networks. in Artificial Intelligence and Pattern Recognition (AIPR), International Conference on. 2016. IEEE.

[21] LeCun, Y., C. Cortes, and C.J. Burges, MNIST handwritten digit database. AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist, 2010.

[22] MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. Available from: http://yann.lecun.com/exdb/mnist/index.html.