Recognition of Handwritten Digits Using Machine Learning Techniques

Mohamed Aboalarbe[1], Hassan Ali[2], Ali Madge [3]

Computer Science Department, Beni-Suef University
Beni-Suef, Egypt

[1]Mohamedaboalarbe@fcis.bsu.edu.eg

[2]Hassanali2596@fcis.bsu.edu.eg

[3]Ali.magde@fcis.bsu.edu.eg

Abstract— Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Given its ubiquity in human transactions, machine recognition of handwriting has practical significance, as in reading handwritten notes in a PDA, in postal addresses on envelopes, in amounts in bank checks, in handwritten fields in forms, etc.

In our paper we aimed to recognize and predict handwritten digits (0:9) based on using Multilayer Perceptron (MLP) Neural Network and PCA.

A combination among two stages of machine learning and image processing is created to build our handwritten digits recognizer system. To get this high accuracy, K-means clustering used in image segmentation and PCA combined with ANN, KNN and Decision tree machine learning algorithms. Also LDA combined with ANN, KNN and Decision tree machine learning algorithms. The high accuracy obtained while using PCA combined with ANN.

It's planned to use deep learning techniques instead of image segmentation algorithms to obtain a high performance in the learning processes.

Keywords— Pattern Recognition, PCA features reduction, LDA features reduction, ANN Classifier, KNN Classifier, Supervised Learning.

I. Introduction

The current renaissance in the field of neural networks is a direct result of the success of various types of deep network in tackling difficult classification and regression problems on large datasets. It may be said to have been initiated by the development of Convolutional Neural Networks (CNN) by LeCun and colleagues in the late 1990s [10]. The aim of this paper is to implement a Multilayer Perceptron (MLP) Neural Network to recognize and predict handwritten digits from 0 to 9. A dataset of 1797 samples was obtained from digits dataset on UCI [7]. Handwritten digit recognition has been a major area of research in the field of Optical Character Recognition (OCR).Based on the input to the system, handwritten digit recognition can be categorized into online and offline recognition. In the online mode, the movements of a pen on a pen-based software screen surface were used to provide input into the system designed to predict the handwritten digits. Meanwhile, the offline mode uses an interface such as a scanner or camera as input to the system [8]. The conversion of an image based on the digit contained to letter codes for further use in a computer or text processing application is the prior step in an off-line handwriting recognition system. This form of data provides a static representation of any handwriting contained. The task of recognizing the handwriting of an individual from another is difficult as each personal possess a unique handwriting style. This is one reason as to why handwriting is considered as one of the main challenging studies. The need for handwritten digit recognition came about the time when combinations of digits were included in records of an individual. The current scenario calls for the need of handwritten digit recognition in banks to identify the digits on a bank cheque and also to collect other user account related information. Moreover, it can be used in post offices to identify pin code box numbers, as well as in pharmacies to identify the doctors' prescriptions. Although there are several image processing techniques designed, the fact that the handwritten digits do not follow any fixed image recognition patterning each of its digits makes it`s a challenging task to design an optimal recognition system. This study concentrates on the offline recognition of digits using an MLP neural network. The difficulty of visual pattern recognition becomes apparent if you attempt to write a computer program to recognize digits. What seems easy when we do it ourselves suddenly becomes extremely difficult. Simple intuitions about how we recognize shapes - "a 9 has a loop at the top, and a vertical stroke in the bottom right" - turn out to be not so simple to express algorithmically. When you try to make such rules precise, you quickly get lost in a morass of exceptions and caveats and special cases. It seems hopeless. Neural networks approach the problem in a different way. The idea is to take a large number of handwritten digits, known as training examples, and then develop a system which can learn from those training examples. In other words, the neural network uses the examples to automatically infer rules for recognizing handwritten digits. Furthermore, by increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy. Many methods have been proposed till date to recognize and predict the handwritten digits. Some of the most interesting are those briefly described below. We made about 9 experiments using different techniques and compare the proposed results to obtain the most accurate one. When we combine PCA technique for features reduction and using MLP ANN the system accuracy was 100%. And here Is a sample images of digits in the used dataset.

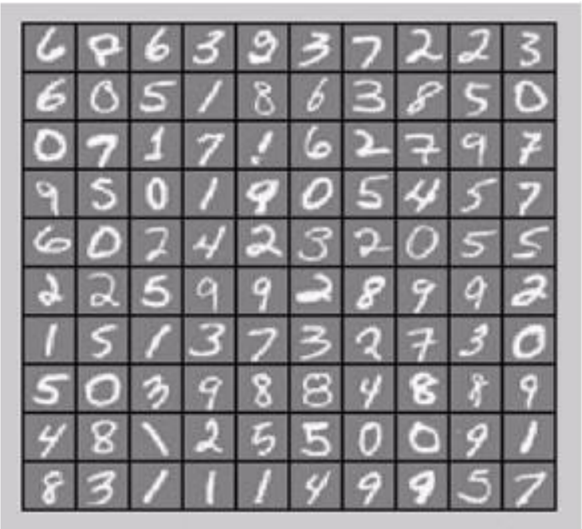


Figure1: Sample handwritten digits from DIGITS

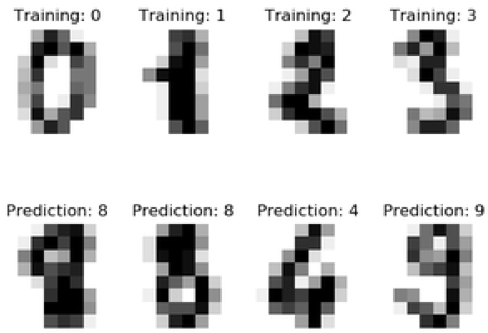
II. RELATED WORK

Some researchers have achieved "near-human performance" on the MNIST database, using a committee of neural networks; in the same paper, the authors achieve performance double that of humans on other recognition tasks. The highest error rate listed on the original website of the database is 12 percent, which is achieved using a simple linear classifier with no pre-processing.
In 2004, a best-case error rate of 0.42 percent was achieved on the database by researchers using a new classifier called the LIRA, which is a neural classifier with three neuron layers based on Rosenblatt's perceptron principles [4].

B. El Kessab, C. Daoui1, B. Bouikhalene, M. Fakir and K. Moro used neural networks (MLP multilayer perceptron) and a method of extraction of characteristics based on the digit form. This work has achieved approximately 80% of success [2]. Clearly it’s not a good result.
Saeed AL-Mansoori used gradient descent back-propagation algorithm for dataset training and the feed-forward algorithm for testing data, also used MLP neural network, sigmoid Activation function for estimating the active sum at the outputs of both hidden and output layers, he reached to 99.32% overall accuracy [1]. That’s a high and very excellent result but with the combination of PCA and ANN we reached to the highest one 100%. Success result. There is no person reached to that result before.
Also, Cireşan, Dan Claudiu, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber has 0.35% error rate in there trial [4].

III. MATERIALS AND METHODS

This paper will handle the identification of the hand write of the digits and train the system on that.



Figur2. Some samples show the error rate while training of the digits from 1 to 9.

To start the work in this paper we collect some information about how to make such that system and we have mentioned in related work.

The system was implemented in python and we have used 2 different platforms (Jupyter and jtbrains pycharm) and all the work were integrated in one platform at the end.

To implement that work we have used some of python libraries and packages like (open CV, scikit learn, matplot, numpy, metrics, nearest neighbours, neural network and tree)

To make the experiment on the dataset we tested it in the pure case, and after that we applied another processes to enhance the results.

The LDA and PCA were applied on the dataset as pre-learning algorithms to get more accuracy.

1- [Assumption 1] using LDA in the pre-learning process to reduce the features of the dataset from 64 to 8 will get a good results and high accuracy of recognition.
2- [Assumption 2] using PCA to the pre-learning process to reduce the features from 64 to 37 will increase the efficiency.

The main 3 classifiers that we used in this research were Neural Network, KNN and Decision tree.

A. Neural Network Architecture

Figure 3 illustrates the architecture of the proposed neural network. It consists of an input layer, hidden layer and an output layer. The input layer and the hidden layer are connected using weights represented as W_{ij}, where i represent the input layer and j represents the hidden layer. Similarly, the weights connecting the hidden and output layer are termed as W_{jk}, where, k represents the output layer. A bias of +1 is included in the neural network architecture for efficient tuning of the network parameters. In this MLP neural network, sigmoid activation function was used for estimating the active sum at the outputs of both hidden and output layers. The sigmoid function is defined as shown in equation 1 and it returns a value within a specified range of [0, 1].

$$g(x) = 1/\ 1+e^{-x} \qquad (1)$$

Here, g(x) represents the sigmoid function and the net value of the weighted sum is denoted as x.

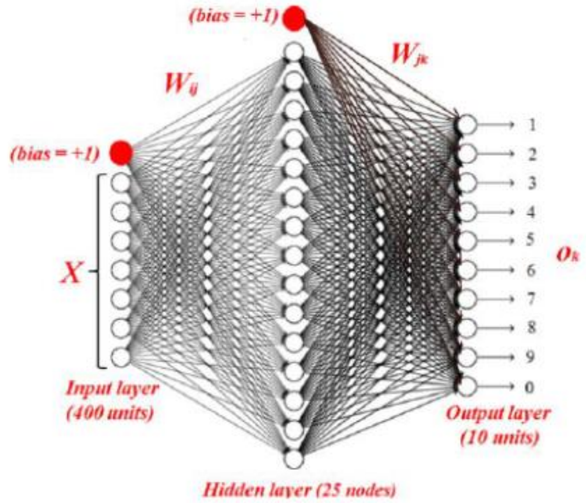


Figure3. The proposed neural network architecture

Input Layer

The 400 pixels extracted from each image is arranged as a single row in the input vector X. Hence, vector X is of size (5000×400) consisting of the pixel values for the entire 5000 samples. This vector X is then given as the input to the input layer. Considering the above mentioned specifications, the input layer in the neural network architecture consists of 400 neurons, with each neuron representing each pixel value of vector X for the entire sample, considering each sample at a time.

Hidden Layer

As per various studies conducted in the field of artificial neural network, there do not exist a fixed formula for determining the number of hidden neurons. Researchers working in this field have however proposed various assumptions to initialize a cross validation method for fixing the hidden neurons [10]. In this study, geometric mean was initially used to find the possible number of hidden neurons. Thereafter, the cross validation technique was applied to estimate the optimal number of hidden neurons. Figure 3 shows a comparative study of the neural network training and testing for 5000 samples with respect to various hidden neurons.

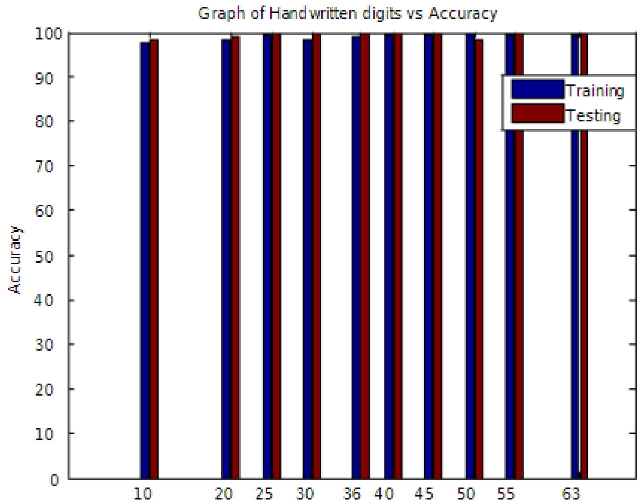


Figure4. Variation of accuracy w.r.t number of hidden neurons.

As observed, a neural network with hidden neuron numbers 63, 55, 45, 40, 36 and 25 produced almost similar accuracies for testing and training. A neural network of hidden neuron number 25 was fixed for further training and testing to reduce the cost during its real time realization.

Output Layer

The targets for the entire 5000 sample dataset were arranged in a vector Y of size (5000×1). Each digit from 0 to 9 was further represented as yk with the neuron giving correct output to be 1 and the remaining as 0. Hence, the output layer consists of 10 neurons representing the 10 digits from 0 to 9.

$$\begin{array}{c}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \\
\begin{array}{cccccccccc}
y_k^{(1)} & y_k^{(2)} & y_k^{(3)} & y_k^{(4)} & y_k^{(5)} & y_k^{(6)} & y_k^{(7)} & y_k^{(8)} & y_k^{(9)} & y_k^{(10)}
\end{array}
\end{array}$$

B. KNN

In pattern recognition, the *k*-nearest neighbors algorithm (*k*-NN) is a non-parametric method used for classification and regression.[1] In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.
- Neighbor distance =

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms.

C. Decision tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values(or expected utility) of competing alternatives are calculated.

A decision tree consists of three types of nodes:

1. Decision nodes – typically represented by squares
2. Chance nodes – typically represented by circles
3. End nodes – typically represented by triangles

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses *Entropy* and *Information Gain* to construct a decision tree.

- Entropy: we need to calculate two types of entropy using frequency tables as follows:

  a) Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

  b) Entropy using the frequency table of two attributes:

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

- Information Gain

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

- Then Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch
- A branch with entropy of 0 is a leaf node & a branch with entropy more than 0 needs further splitting.
- The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

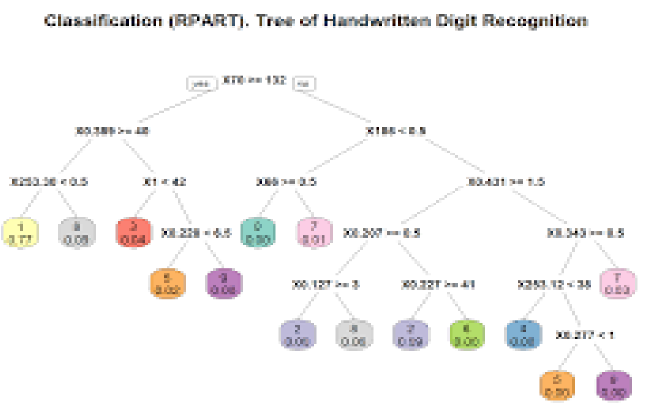


Figure5.Classification Decision tree of handwritten digits.

By comparing the results with Saeed Al-mansoori in his paper [9] according to the neural network results we will make a high score over him because the results in Saeed's excrement were 93932% as a final accuracy bet we get the result = 100% as a final accuracy

IV. RESULTS AND DISCUSSION

V. ALL OF THE FOWLING RESULTS WERE APPLIED ON THE DIGITS DATASET AND GET THIS RESULTS BENCHMARK:

VI. 1- BY APPLYING THE KNN, DECISION TREE AND NURAL NETWORK ON THE PURE DATASET THE RESULTS NOTICED AS

Algorithm	KNN	DT	NN
Accuracy	85.3%	98.8%	100%

VII. FIGURE6. TABLE SHOW THE RESULTS OF THE KNN, DT AND NN ON THE PURE DATASET

VIII. 2- BY USING THE LDA FOR FEATURE REDUCTION AND REDUCE THE FEATURE OF THE DATASET FROM 64 FEATURE TO 8 FEATURE, THE RESULT WERE:

Algorithm	LDA with KNN	LDA with DT	LDA with NN
Accuracy	97.8%	100%	99.1%

IX. FIGURE7. TABLE SHOWS THE RESULTS OF APPLYING LDA BEFORE EACH ALGORITHM.

X. 3- IN THE LAST TIME WE APPLIED PCA ALGORITHM FOR FEATURE REDUCTION AND THE ALGORITHM REDUCED THE FEATURE FROM 64 TO 37 FEATURE, AND WE GET THIS RESULTS:

Algorithm	PCA with KNN	PCA with DT	PCA with NN
Accuracy	99.9%	100%	100%

XI. FIGURE8. TABLE SHOW THE RESULTS OF APPLYING PCA BEFORE EACH ALGORITHM.

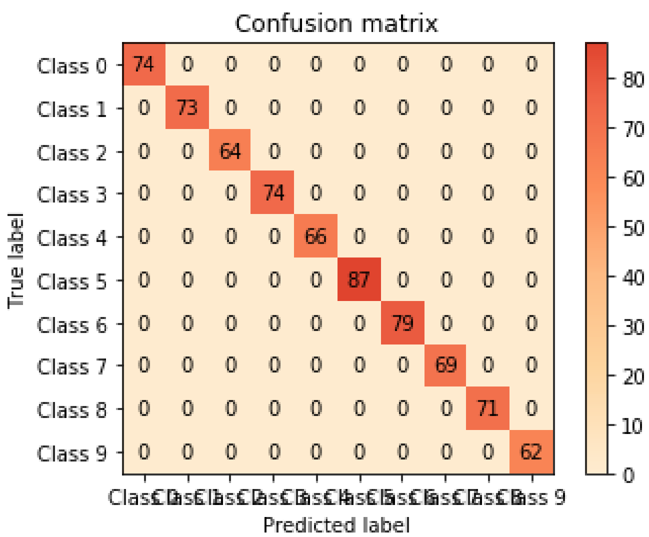


FIGURE9.Confusion matrix of the 9 classes that show the accuracy (100%)

V .CONCLUSION AND FUTURE WORK

In this paper, a Multilayer Perceptron (MLP) Neural Network was implemented to address the handwritten digit recognition problem. The proposed neural network was trained and tested on a dataset attained from DIGITS. The system performance was observed by the combination of PCA technique with MLP ANN algorithm the system was found to provide the optimal parameters to the problem. The proposed system was proved efficient with an overall testing accuracy of 100%. We aim to develop a mathematical model which can use deep learning techniques and get more training samples to generalize the model for all numbers and alphabet {A..Z}.

REFERENCES

[1] Saeed AM. Intelligent Handwritten Digit Recognition using Artificial Neural Network.May 2015.

[2] El Kessab B, Daoui C, Bouikhalene B, Fakir M, Moro K. Extraction method of handwritten digit recognition tested on the mnist database. International Journal of Advanced Science & Technology. 2013 Jan;50.

[3] LeCun, Yann, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. "Handwritten digit recognition with a back-propagation network." In Advances in neural information processing systems, pp. 396-404. 1990.

[4] https://en.wikipedia.org/wiki/MNIST_database.

[5] Lee, Yuchun. "Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks." Neural computation 3, no. 3 (1991): 440-449.

[6] C. Kaynak (1995) Methods of Combining Multiple Classifiers and Their Applications to Handwritten Digit Recognition, MSc Thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University.

[7] http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of +Handwritten+Digits.

[8] R. Plamondon and S. N. Srihari, On-line and off-line handwritten character recognition: A comprehensive "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland. Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no. 1, 2000, 63-84.

[9] AL-Mansoori, Saeed. (2015). Intelligent Handwritten Digit Recognition using Artificial Neural Network. . 10.13140/RG.2.1.2466.0649.

[10] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE.1998; 86:2278–2324.

[11] Le Cun, Y., Jackel, L.D., Boser, B., Denker, J.S., Graf, H.P., Guyon, I., Henderson, D., Howard, R.E. and Hubbard, W., 1989. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, *27*(11), pp.41-46.

[12] Revow M, Williams CK, Hinton GE. Using generative models for handwritten digit recognition. IEEE transactions on pattern analysis and machine intelligence. 1996 Jun; 18(6):592-606.

[13] Cireşan, Dan Claudiu, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. "Deep, big, simple neural nets for handwritten digit recognition." Neural computation 22, no. 12 (2010): 3207-3220.