A Comparative Study on Handwritten Digit Recognizer using Machine Learning Technique

[1]R. Kanniga Devi
Associate Professor, Department of Computer Science and Engineering, School of Computing
Kalasalingam Academy of Research and Education,
Krishnankoil, Tamilnadu, India rkannigadevi@gmail.com

[2]G. Elizabeth Rani
Assistant Professor, Department of Computer Science and Engineering, School of Computing
Kalasalingam Academy of Research and Education,
Krishnankoil, Tamilnadu, India g.elizabeth@klu.ac.in

***Abstract*—Ability for accurate digit recognizer modelling and prediction is critical for pattern recognition and security. A variety of classification machine learning algorithms are known to be effective for digit recognition. The purpose of this experiment is rapid assessment of multiple types of classification models on digit recognition problem. The work offers an environment for comparing four types of classification models in a unified experiment: Multiclass decision forest, Multiclass decision jungle, Multiclass Neural Network and Multiclass Logistic Regression. The work presents assessment results using 6 performance metrics: Overall accuracy, Average accuracy, Micro-averaged precision, Macro-averaged precision, Micro-averaged recall and Macro-averaged recall. The experimental results showed that the highest accuracy was obtained by Multiclass Neural Network with the value of 97.14%. The operational tool has been published to the web and openly available at Azure Machine Learning Studio for experiments and extensions.**

***Keywords*—Digit recognizer, Machine Learning, Classification Algorithms, Azure Machine Learning Studio.**

I. INTRODUCTION

Digit recognition problem is a promising problem in handwriting recognition problem. It is also one of the challenging problems in computer vision and machine learning [1]. It is so called challenging task as developing an accurate automated recognition of handwritten digits is difficult. The applications of digit recognition includes bank check processing and online form data submitted by users by using digital devices [2]. The variation in handwriting among people makes it difficult to train the computers to recognize handwritten digits. This work employs machine learning techniques to address the digit recognition problem. To analyze various classification techniques and finding the best technique for digit recognition, the work focuses on utilizing a tool that would embrace multiple types of classification models with diverse performance metrics.

The paper is structured as follows. The related work is discussed in section 2. Section 3 elaborates the proposed work and experimental platform. Experimental results are provided in Section 4. Section 6 concludes with the summary and scope for future work.

II. RELATED WORK

Many researchers applied machine learning techniques to solve the handwritten digit recognition problem.

Hayder [3] work applied *K*-Nearest Neighbor(*KNN*), Neural Networks (*NN*) and Decision Tree (*DT*) for digit classification. The *KNN* assigned specific class data to those that possess the most representatives within the nearest neighbors of the class. The Decision Tree classifier used Hunt's algorithm to construct a tree model that can classify the dataset of handwritten digits using class-based rules. This work also uses a fully connected Artificial Neural Network (*ANN)* with one and two hidden layer on 28*28 image sizes. The work considered Mixed National Institute of Standards and Technology (MNIST) data set and the results showed that *NN* achieved high accuracy.

Meiyin Wu [4] work applied Deep Belief NetWork (*DBN*) and Convolutional Neural Network (*CNN*) which are deep learning techniques to the handwritten character recognition problem. this work considered both the real-world handwritten character database and MNIST database and the results showed that the accuracy obtained by *CNN* was greater than *DBN*.

Chayaporn Kaensar [5] work conducted a study on digit recognition using different classifiers The work considered dataset from *UCI* repository and used *WEKA* tool kit to train and test the dataset. The results showed that Support Vector Machine (*SVM*) classified accurately than other classifiers, but *SVM* consumed lot of training time.

Raid Saabni [6] work applied deep neural network for handwritten digit recognition. To enable high recognition and achieve reasonable training time, this work used multi layer neural network trained by deep network and back propagation and structure. Pre-trained layers with sparse auto encoders were used to train the neural network to achieve high accuracy rates. The work considered the datasets namely, MNIST, ORAND−CAR, CV L and real-world bank checks.

Adriano Mendes Gil [7] work used SVM binary classifiers for handwritten digit recognition. This work considered input in two ways namely, digit characteristics and the whole images. This work provided used binary classifiers to form a structure for a multiclass classifier. This work was performed on MNIST database and achieved greater accuracy.

Faizan Farooq [8] work applied artificial neural networks and logistic regression to recognize handwritten digits from 0 to 9. To perform multiclass classification, this work used artificial neural networks and One-vs-All logistic regression. They applied vectorization which is an advanced optimization algorithm to make the performance faster. This work considered a database of five thousand handwritten digits and reported that the system is very effective in recognizing the digit.

From the study of the related works, it is observed that there exists multiple machine learning approaches for hand written digit recognition problem. However, authors have performed limited set of comparison as developing code to test on each classifier is a time consuming task.

978-1-7281-4407-8/19/$31.00 ©2019 IEEE

Hence, the proposed work simplifies the task for comparison with a open source tool and considers four types of multi class classifiers for the digit recognition problem. The multiclass classifiers used in the proposed work are published as a web service, which can be used by researchers to conduct experiments with their own datasets.

III. Proposed Work

The objective of the work is to accurately recognize digits from handwritten images dataset by using various types of classification machine learning models namely, Multiclass decision forest, Multiclass decision jungle, Multiclass Neural Network and Multiclass Logistic Regression and identify a model with higher performance.

A. Classification algorithms description

This work experiments with the following four classification algorithms: *Multiclass Decision Forest:* Multiclass Decision Forest uses the decision forest algorithm to build multiple decision trees and applies voting approach to identify the most accepted output class. *Multiclass Decision Jungle:* Multiclass Decision Jungle are the extension to decision forest which uses decision jungle algorithm to create a Directed Acyclic Graphs (DAGs)-based multiclass classification model. *Multiclass Neural Network:* Multiclass Neural Network uses neural network algorithm to create a multiclass classification model which predicts a target with multiple values. *Multiclass Logistic Regression:* Multiclass Logistic Regression fits data to a logistic function and predicts target with multiple values.

B. Experimental Platform

Microsoft Azure Machine Learning Studio (MLS) [9] was selected for conducting experiments. MLS contributed to the objective of the study with the features such as it is a free Cloud-based machine learning as a service, hence no need to set up and maintain software. It has ready to use built-in classification models namely, Multiclass decision forest, Multiclass decision jungle, Multiclass Neural Network and Multiclass Logistic Regression. Users can publish results as web service and reuse published experiments. The configuration settings of all the four classification algorithm is presented from Table I to IV.

TABLE I CONFIGURATION SETTINGS OF MULTICLASS DECISION FOREST

Multiclass Decision Forest Properties	Value
Resampling method	Bagging
Trainer Model	Single parameter
Number of Trees	8
Maximum Depth	32
Random Splits pers Node	128
Samples per leaf node	1

The Multiclass Decision Forest configuration presents the resampling method using which individual trees can be created by using Bagging method. The Bagging method creates each tree on a new sample. The trainer model uses the single parameter as we know how to configure the model and supply set of values as arguments. The maximum number of trees to be created is also mentioned. The value 8 shows the limit on the maximum depth of the decision tree. The split value specifies the maximum number of splits while creating each node of the tree. The samples per leaf node indicates the minimum number of conditions to create leaf node.

TABLE II CONFIGURATION SETTINGS OF MULTICLASS DECISION JUNGLE MODEL

Multiclass Decision Jungle Properties	Value
Resampling method	Bagging
Trainer Model	Single parameter
Decision DAGs	8
Depth of the DAGs	32
Width of the DAGs	128
Optimization steps	2048

The Multiclass Decision Jungle configuration presents the resampling method using which individual trees can be created by using Bagging method. The Bagging method creates each tree on a new sample. The trainer model uses the single parameter as we know how to configure the model and supply set of values as arguments. The value of decision DAGs indicates the maximum number of graphs to be created. The depth and width of the DAGs indicates the maximum depth and width of each graph. The optimization steps states the number of iterations over the data while building each DAG.

TABLE III CONFIGURATION SETTINGS OF MULTICLASS NEURAL NETWORK

Multiclass Neural Network Properties	Value
Trainer mode	Single Parameter
Hidden layer	Fully-connected
Hidden nodes	100
Learning rate	0.1
Learning iterations	100
Learning weights diameter	0.1
Momentum	0
Normalizer	Min-Max normalizer
Random number seed	Null

The Multiclass Neural Network configuration presents the trainer model which uses the single parameter as we know how to configure the model and supply set of values as arguments. the fully connected case in hidden layer indicates that the output layer is fully connected to hidden layer and the hidden layer is fully connected to the input layer. The number of hidden nodes is set to 100. The learning rate specifies the size of the step taken at each iteration and the learning iterations is set to 100. The initial node weight is specified by learning weights diameter and the momentum is set to 0. The Min-Max normalizer rescales every feature to (0,1) interval. The random number seed is set to null as it does not want to ensure repeatability.

TABLE IV CONFIGURATION SETTINGS OF MULTICLASS LOGISTIC REGRESSION

Multiclass Logistic Regression Properties	Value
Trainer mode	Single Parameter
Optimization tolerance	1E-07
L1 weight	1
L2 weight	1
L-BFGS Memory size	20
Random seed	Null

The Multiclass Logistic Regression configuration presents the trainer model which uses the single parameter as we know how to configure the model and supply set of values as arguments. The optimization tolerance specifies the threshold value for optimizer convergence. Regularization is a method for preventing overfitting by penalizing models and hence L1 weight and L2 weight are set to 1. L-BFGS stands for limited memory Broyden-Fletcher-Goldfarb-Shanno, which specifies the amount of

memory to use for optimization. The random number seed is set to null as it does not want to ensure repeatability.

In the experiments, the MLS standard performance metrics were used to evaluate models using the confusion matrix: Accuracy, precision and recall. Accuracy is the ratio of correct results to total cases. Precision is the ratio of correct results over all positive results. Recall is the part of all true results returned. This work also considers micro and macro average on precision and recall. The Micro-average method sums up the True Positives (TP), False Positives(FP) and False Negatives(FN) of the model. The average precision and recall of the system using the Micro-average method is calculated as in Equation 1 and 2:

Micro-average of precision =

$$\frac{(TP)}{(TP + FP)} \tag{1}$$

Micro-average of recall =

$$\frac{(TP)}{(TP + FN)} \tag{2}$$

IV. Experimental Results

Experiment with Azure built-in models workflow chart is presented in Fig. 1- 8.

A. Dataset Description

The benchmarked dataset of handwritten images is considered from Modified National Institute of Standards and Technology (MNIST) in order to perform experiments using various classification algorithms. The gray-scale image dataset contain digit from zero to nine. Each image is of size 28*28 pixels which makes a total of 784 pixels. Each pixel value indicates the lightness or darkness of the image. This pixel-value is an integer between 0 (with lower numbers meaning lighter) and 255 (with higher numbers meaning darker). In total, the data set is made up of 785 columns, of which the first column is the label, which means the digit drawn by the user. The total number of rows in the data set is 4,200, of which 60% is set for training dataset and 40% is set for testing data.

Experiment-1:

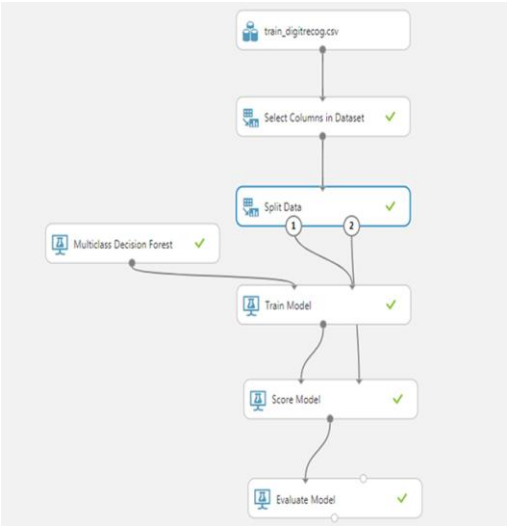

Fig 1 Visual interface for multiclass decision forest experiment

Fig 2 Confusion matrix for multiclass decision forest

The data set is input in csv format, in which all the fields of the data set is selected to analysis. The dataset is split in such a way that 60% of the dataset is used for training the model and 40% of the dataset is used for testing the model. the classifier used in Fig 1 is Multi Class Decision Forest and scoring and evalaution of model is done to visualize the results in confusion matrix format. The confusion matrix shows the performance of the model by summarizing the correct and incorrect predictions with count values. From this, accuracy of prediction can be calculated.

Experiment-2:

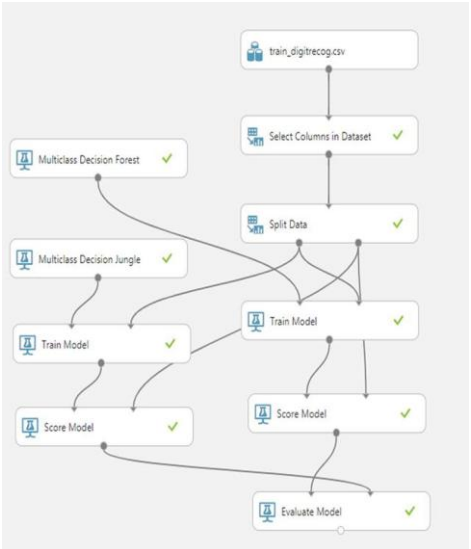

Fig 3 Visual interface for multiclass decision jungle experiment

Fig 4 Confusion matrix for multiclass decision jungle

The data set is input in csv format, in which all the fields of the data set is selected to analysis. The dataset is split in such a way that 60% of the dataset is used for training the model and 40% of the dataset is used for testing the model. the classifier used in Fig 1 is Multi Class Decision Jungle and scoring and evalaution of model is done to visualize the results in confusion matrix format. The confusion matrix shows the performance of the model by summarizing the correct and incorrect predictions with count values. From this, accuracy of prediction can be calculated.

Experiment-3:

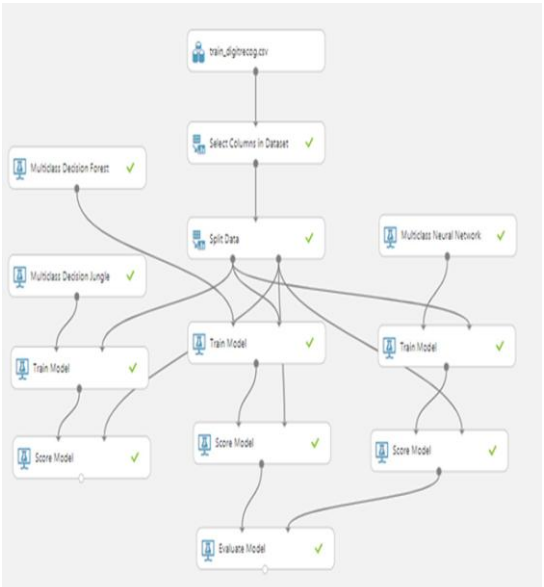

Fig 5 Visual interface for multiclass neural network experiment

Fig 6 Confusion matrix for multiclass neural network

The data set is input in csv format, in which all the fields of the data set is selected to analysis. The dataset is split in such a way that 60% of the dataset is used for training the model and 40% of the dataset is used for testing the model. the classifier used in Fig 1 is Multi Class Neural Network and scoring and evalaution of model is done to visualize the results in confusion matrix format. The confusion matrix shows the performance of the model by summarizing the correct and incorrect predictions with count values. From this, accuracy of prediction can be calculated.

Experiment-4:

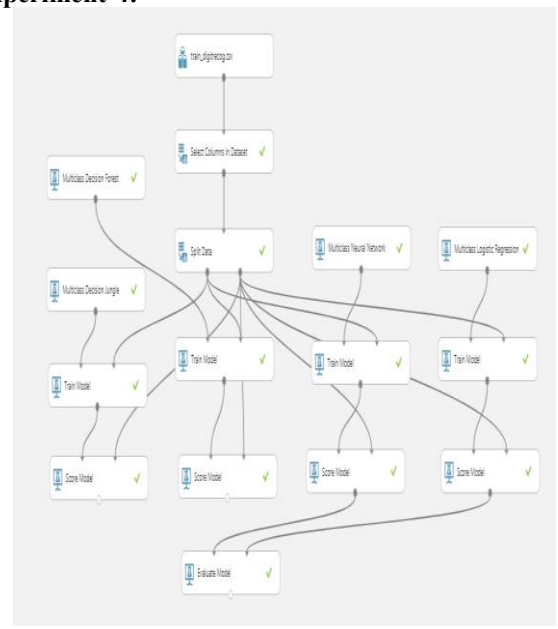

Fig 7. Visual interface for multiclass logistic regression experiment

Fig 8 Confusion matrix for multiclass logistic regression

The data set is input in csv format, in which all the fields of the data set is selected to analysis. The dataset is split in such a way that 60% of the dataset is used for training the model and 40% of the dataset is used for testing the model. the classifier used in Fig 1 is Multi Class Logistic Regression and scoring and evalaution of model is done to visualize the results in confusion matrix format. The confusion matrix shows the performance of the model by summarizing the correct and incorrect predictions with count values. From this, accuracy of prediction can be calculated.

B. Performance Metrics

In the experiments 1 to 4, the Macro-average method for precision and recall takes the average of the precision and recall of the model on different sets. Macro-average method can be used to understand how the system performs overall across different sets of data whereas, Micro-average can be used when dataset varies in size. The average accuracy is the average of each accuracy per class and the overall accuracy is the ratio of correctly predicted items and total items.

From the results presented in Table V and VI, it is observed that Multiclass Neural Network outperforms the other classifiers.

TABLE V. MEASURES OF CLASSIFIERS

Classification Algorithm	Overall accuracy	Average accuracy	Micro-averaged precision
Multiclass Decision Forest	0.93	0.98	0.93
Multiclass Decision Jungle	0.87	0.97	0.87
Multiclass Neural Network	0.97	0.99	0.97
Multiclass Logistic Regression	0.94	0.98	0.94

TABLE VI MEASURES OF CLASSIFIERS

Classification Algorithm	Macro-averaged precision	Micro-averaged recall	Macro-averaged recall
Multiclass Decision Forest	0.93	0.93	0.93
Multiclass Decision Jungle	0.87	0.87	0.87
Multiclass Neural Network	0.97	0.97	0.97
Multiclass Logistic Regression	0.94	0.94	0.94

V. CONCLUDING REMARKS

The contribution of this study is delivering a web service based on Microsoft Azure Machine Learning Studio for rapid assessment of multiple types of multiclass classification models on digit recognition problem. The higher performance was demonstrated by Multiclass Neural Network model. The web service has been published at Azure MLS (KannigaDevi2019). It can be further used by researchers to conduct experiments with their own data sets or add some more classification models to the web service.

As part of the future work, this work tries to improve the handwritten digits classification accuracy further to 100%.

REFERENCES

[1] Caiyun Ma, Hong Zhang, Effective Handwritten Digit Recognition Based on Multi-feature Extraction and Deep Analysis, IEEE 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015, 297-301.

[2] Mahmoud M. Abu Ghosh, Ashraf Y. Maghari, A Comparative Study on Handwriting Digit Recognition Using Neural Networks, IEEE International Conference on Promising Electronic Technologies, 77-81, 2017.

[3] Hayder Naser Khraibet AL-Behadili, Classification algorithms for determining handwritten digit, Iraq Journal of Electrical and Electronic Engineering, Vol.12, No.1, 2016, 96-102.

[4] Meiyin Wu, Li Chen, Image recognition based on deep learning, IEEE Chinese Automation Congress (CAC), 2015, 542-546.

[5] Chayaporn Kaensar, A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor, Proceedings of the 9th International Conference on Computing and Information Technology (IC2IT2013), Springer, 2013, 155-163.

[6] Raid Saabni, Recognizing handwritten single digits and digit strings using deep architecture of neural networks, Proceedings of the IEEE International Conference on Artificial Intelligence and Pattern Recognition (AIPR),2016, 26-31.

[7] Adriano Mendes Gil, Cícero Ferreira Fernandes Costa Filho, Marly Guimarães Fernandes Costa, Handwritten Digit Recognition Using SVM Binary Classifiers and Unbalanced Decision Trees. Proceedings of International Conference Image Analysis and Recognition (ICIAR 2014). Lecture Notes in Computer Science, vol 8814. Springer, Cham, 2014, 246-255.

[8] Faizan Farooq, Siddhant Tandon, Pankaj Parashar and Prateek Sengar, Vectorized Code Implementation of Logistic Regression and Artificial Neural Networks to Recognize Handwritten Digit, Proceedings of the 1st IEEE International Conference on Power Electronics. Intelligent Control and Energy Systems (ICPEICES-2016), 2016, 1-5.

[9] [online] https://studio.azureml.net, Last accessed May 2019.