# Offline Handwritten Digits Recognition Using Machine learning

**Conference Paper** · October 2018

**4 authors:**

Lee J. Wells
Western Michigan University
**55** PUBLICATIONS **1,051** CITATIONS
SEE PROFILE

Shengfeng Chen
Western Michigan University
**6** PUBLICATIONS **62** CITATIONS
SEE PROFILE

Rabia Emhamd Al Mamlook
Western Michigan University
**65** PUBLICATIONS **675** CITATIONS
SEE PROFILE

Yuwen Gu
Western Michigan University
**4** PUBLICATIONS **22** CITATIONS
SEE PROFILE

# Offline Handwritten Digits Recognition Using Machine learning

Shengfeng Chen
[1]Department of Industrial Engineering Western Michigan University, Kalamazoo, MI.
shengfeng.chen@wmich.edu

Rabia Almamlook
[2]Department of Industrial Engineering Western Michigan University, Kalamazoo, MI.
rabiaemhamedm.almamlook@gwmich.edu

Yuwen Gu
[3]Department of Industrial Engineering Western Michigan University, Kalamazoo, MI.
yuwen.gu@wmich.edu

Dr. Lee wells
[4]Department of Industrial Engineering Western Michigan University, Kalamazoo, MI.

lee.wells@wmich.edu

## ABSTRACT

The problem of handwritten digit recognition has long been an open problem in the field of pattern classification. Several studied have shown that Neural Network has a great performance in data classification. The main objective of this paper is to provide efficient and reliable techniques for recognition of handwritten numerals by comparing various existing classification models. This paper compares the performance of five machine learning classifier models namely Neural Network, K-Nearest Neighbor (K-NN), Random Forest, Decision Tree and Bagging with gradient boost. Results indicate that K-NN classifier outperform Neural Network with significant improved computational efficiency without sacrificing performance. They both outperformed the other classifiers: Random Forest, Decision Tree and Bagging with gradient boost. We also discovered that as the training data is increasing the accuracy of the classifier is also improved. The result of this paper shows that K-NN has equally high accuracy of 96.7% compared to Neural Network of 96.8%, but K-NN achieves a processing speed with almost 10 times faster. The analysis presented in this paper suggests that the K-NN combined with preprocessing methods is capable of achieving great performance apart from Neural Network when used as a classification algorithm in offline handwritten digit recognition.

***Keywords:*** *Handwritten character recognition (HCR), classification- K-Nearest Neighbor*

## 1. Introduction

The problem of handwritten numerals recognition has been widely studied in recent years and the large quantity of preprocessing methods and classification algorithms have been developed. (Wang & Yan, June 2000). However, handwritten numerals recognition is still a challenge for us. The main difficulty of handwritten numerals recognition is the serious variance in size, translation, stroke thickness, rotation and deformation of the numeral image because of handwritten digits are written by different users and their writing style is different from one user to another user. It is even impossible to construct a database, including all the typical samples of the unconstrained numerals. For feature extraction of character recognition, various approaches have been proposed (O.D. Trier, A.K. Jain, & T. Taxt, 1995). Handwritten digit recognition is an active area of research in optical character recognition applications and pattern classifications (Tuan,2002). Handwriting recognition is classified as offline handwriting recognition and online handwriting recognition (Salvador et Al 2014). If handwriting is recognized while writing through touchpad using the stylus pen, it's called online handwriting recognition. In this case, handwriting is scanned and then understood by the computer, it is called offline handwriting recognition

Several studied have used different ways to handwritten digit with different machine learning techniques Khotanzad et at (1998) who have applied the concepts of Machine Learning and Neural Networks to recognize and determine the handwritten digits from its image. This study has shown that digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques like deep learning. The finding is similar result with different machine learning techniques to Maureen et al (2017) have been employ Multiple classifiers to discriminate male and female writings including artificial neural networks (ANN), support vector machine (SVM), nearest neighbor classifier (NN), decision trees (DT) and random forests (RF). This study showed that neural networks (ANN) is the best performance. This finding is different from Gorgevik and Cakmakov (2017) who have developed Support Vector Machine (SVM) based digits recognition system for handwritten Roman numerals. Perwej et al (2012) have provided the conversion of handwritten data into electronic data, nature of handwritten characters and the neural network approach to form machine competent of recognizing hand-written characters while as Liu et al (2013) discourses a comprehensive criterion of handwritten digit recognition with numerous state of the art approaches, feature illustrations, and datasets. Though, the relationship of training set size versus accuracy and the dataset-independence of the trained models are studied. Le Cun et al (1998) present convolution neural networks into the handwritten digit recognition research and describes a system which can still be considered state of the art.

This study focuses on feature extraction and classification. The performance of a classifier can rely as much on the quality of the features as on the classifier itself. Moreover, our study presents an efficient offline handwritten character recognition system based on diagonal features and transitions features using the k-NN classifier. Diagonal and transitions features of a character have been computed based on the distribution of points on the bitmap image of the character. In this study, we compare the performance of five different machine learning classifiers for recognition of digits. The five classifiers namely neural network, K-Nearest Neighbor, Random Forest, Decision Tree and Bagging with gradient boost. The combination of different classifiers, such as ANN, and K-NN implanted in a recognition system is also a suitable method of increasing recognition performance, because different classifier combinations have different merits in dealing with discriminant problems. This study train and test K-Nearest Neighbors classifier in RapidMiner for pattern analysis in solving handwritten digit recognition problems using the MNIST database. Although K-NN is a lazy learner algorithm, it's selected because of its high accuracy in digits predicting and training efficiency when compared to other operators like Neural Net, random forest etc. Here, the preprocessing of the dataset is done using MATLAB. The main purpose of this thesis is to build a reliable method for the recognition of handwritten digit strings. In order to accomplish the recognition task, first, the digit string is segmented into individual digits. Then, a digit recognition module is employed to classify each segmented digit completing the handwritten digit string recognition task. In this study, a novel method, which uses the K-Nearest Neighbors classifier, is proposed to achieve high performance on the digit string segmentation problem. Our main contribution in this work is that KNN was more performance than Neural Net to discover that the result of our study is different than other studies. K-Nearest Neighbor (k-NN) classifier performed slightly better than the neural network.

## 2. Research Methodology

### 2.1 Description of the dataset

In this paper, we used the MNIST database consisting of offline handwritten digits ranging from 0-9. The database was constructed from Special Database 3 (SD-3) and Special Database 1 (SD-1) that contain binary images of handwritten digits. SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students. For the results to be independent of both datasets, MNIST dataset was built by mixing NIST SD-1 and SD-3. The total number of digit image samples (70,000), the total number for training (60,000) and testing (10,000), and the subtotal number for each digit are shown in table 1. Each digit is a gray-level fixed-size image with a size of 28 x 28 (or 784 pixels) in total as the features.

**Table 1.** Dataset

| Digits | # Training | # Testing | Subtotal |
|---|---|---|---|
| 9 | 5949 | 1009 | 6958 |
| 8 | 5851 | 974 | 6825 |
| 7 | 6265 | 1028 | 7293 |
| 6 | 5918 | 958 | 6876 |
| 5 | 5421 | 892 | 6313 |
| 4 | 5842 | 982 | 6824 |
| 3 | 6131 | 1010 | 7141 |
| 2 | 5958 | 1032 | 6990 |
| 1 | 6742 | 1135 | 7877 |
| 0 | 5923 | 980 | 6903 |
| Total | 60,000 | 10,000 | 70,000 |

The dataset contained examples from approximately 250 writers. The sets of writers for the dataset were disjoint. The Sample digit database of MNIST is shown in figure 1.

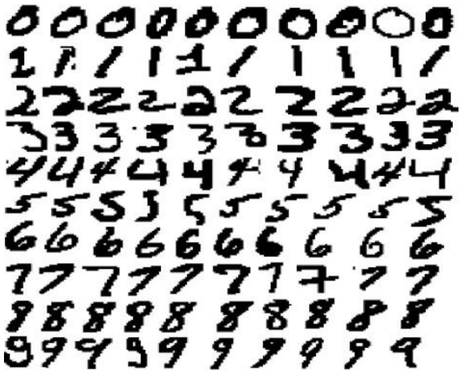


**Figure 1**. Sample images of MNIST digit database

## 2.2 Machine learning method

### 2.2.1 K-Nearest-Neighbor method

The K-Nearest-Neighbor is a type of Lazy Learners and one of the most commonly used nearest neighbor-based algorithms, works on learning by analogy, that is by comparing a given test example with training example that are similar to it. The training examples are described by n attributes. Each example represents a point in a n-dimensional space. In this way, all the training examples are stored in a n-dimensional pattern space. When given an unknown example, a k-nearest-neighbor classifier searches the pattern space for the k training example that are closest to the unknown example. These k training examples are the k “nearest neighbors” of the unknown example. The closeness that is the Euclidean distance between two examples, X1 = (x11, x12…, x1n) and X2 = (x21, x22…, x2n) is given by

$$\boldsymbol{dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{21})^2}}$$

Before using the equation above, each attribute is normalized as it helps to prevent attributes with initially large ranges from outweighed attributes with initially small ranges. This can be done by Min – max normalization, for transforming a value v of a numeric attribute A to v’ in the range [0,1] by computing:

$$v' = \frac{v - min_A}{max_A - min_A}$$

where $min_A$ and $max_A$ are the minimum and maximum values of attribute A.

### 2.2.2 Artificial neural network (ANN)

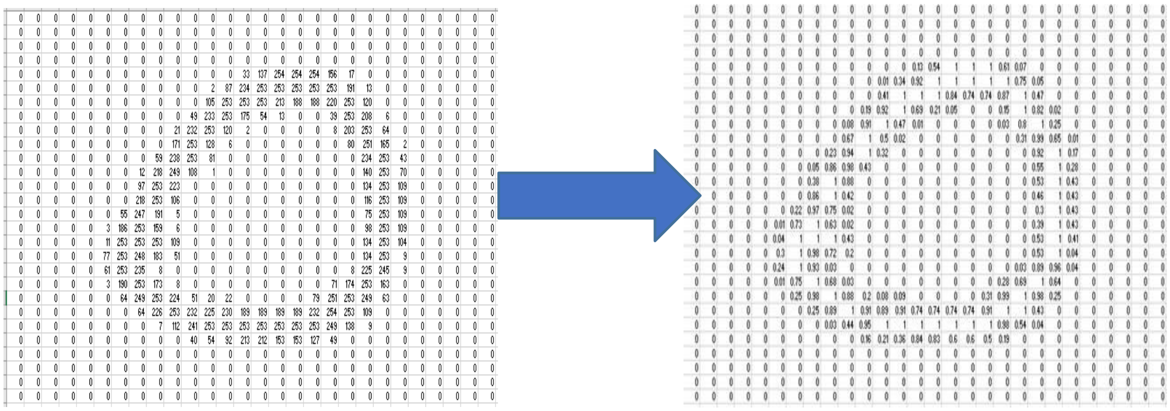
ANNs are non-linear mapping structure. ANNs can recognize correlated patterns between input data set and corresponding target values. ANNs has huge capacity in prediction, pattern recognition, data compression, decision-making, etc. ANNs are recently used in the classification problem where regression model and other statistical techniques have traditionally been applied (Paliwal et at,2009 and Chukwu and K.A. Adepoju,2012). Today, there are many different models of ANNs. The differences might be the topology, the functions, the hybrid models, the accepted values, the learning algorithms, etc.

## 2.3. Preprocessing Recognition System.

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 and they are discussed as follows.

### 2.3.1 Normalization

The first step in data preprocessing is data normalization. This is done to apply distance calculations on it. This involves transforming the data to fall within a smaller or common range, such as [0, 1]. The raw image data is based on the standard 8-bit unsigned integer which has a high value range of [0, 255] at each pixel (attribute). Expressing an attribute in smaller units will lead to a larger range for that attribute, thus tend to give such attributes greater effect or "weight." Normalizing the data attempts to give all attributes an equal weight and the input values for each attribute measured in the training tuples. It will help speed up the learning phase and it makes possible to add a new attribute to the dataset in the later stage as long as the new attribute is also normalized. Figure 2 shows the normalization effect on the handwritten image. The Raw image is in the range of [0, 255]. After normalizing it is reduced to the range of [0, 1].



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 | 137 | 254 | 254 | 254 | 156 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 87 | 234 | 253 | 253 | 253 | 253 | 253 | 191 | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 105 | 253 | 253 | 253 | 213 | 188 | 188 | 220 | 253 | 120 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 49 | 233 | 253 | 175 | 54 | 13 | 0 | 0 | 39 | 253 | 208 | 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 232 | 253 | 120 | 2 | 0 | 0 | 0 | 0 | 8 | 203 | 253 | 64 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 171 | 253 | 128 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 80 | 251 | 165 | 2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 59 | 238 | 253 | 81 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 234 | 253 | 43 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 218 | 249 | 108 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 140 | 253 | 70 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 97 | 253 | 223 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 134 | 253 | 109 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 218 | 253 | 106 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 116 | 253 | 109 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 55 | 247 | 191 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 75 | 253 | 109 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 3 | 186 | 253 | 159 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 98 | 253 | 109 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 11 | 253 | 253 | 253 | 109 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 134 | 253 | 104 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 77 | 253 | 248 | 183 | 51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 134 | 253 | 9 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 61 | 253 | 235 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 225 | 245 | 9 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 3 | 190 | 253 | 173 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 71 | 174 | 253 | 163 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 64 | 249 | 253 | 224 | 51 | 20 | 22 | 0 | 0 | 0 | 0 | 79 | 251 | 253 | 249 | 63 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 64 | 226 | 253 | 232 | 225 | 230 | 189 | 189 | 189 | 189 | 232 | 254 | 253 | 109 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 112 | 241 | 253 | 253 | 253 | 253 | 253 | 253 | 253 | 249 | 138 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 54 | 92 | 213 | 212 | 153 | 153 | 127 | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.13 | 0.54 | 1 | 1 | 1 | 0.61 | 0.07 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.01 | 0.34 | 0.92 | 1 | 1 | 1 | 1 | 1 | 0.75 | 0.05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.41 | 1 | 1 | 1 | 0.84 | 0.74 | 0.74 | 0.87 | 1 | 0.47 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.19 | 0.92 | 1 | 0.69 | 0.21 | 0.05 | 0 | 0 | 0.15 | 1 | 0.82 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.08 | 0.91 | 1 | 0.47 | 0.01 | 0 | 0 | 0 | 0 | 0.03 | 0.8 | 1 | 0.25 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.67 | 1 | 0.5 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 0.31 | 0.99 | 0.65 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.23 | 0.94 | 1 | 0.32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.92 | 1 | 0.17 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.05 | 0.86 | 0.98 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.55 | 1 | 0.28 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.38 | 1 | 0.88 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.53 | 1 | 0.43 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0.86 | 1 | 0.42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.46 | 1 | 0.43 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.22 | 0.97 | 0.75 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.3 | 1 | 0.43 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.01 | 0.73 | 1 | 0.63 | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.39 | 1 | 0.43 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.04 | 1 | 1 | 1 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.53 | 1 | 0.41 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.3 | 1 | 0.98 | 0.72 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.53 | 1 | 0.04 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.24 | 1 | 0.93 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0.89 | 0.96 | 0.04 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.01 | 0.75 | 1 | 0.68 | 0.03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.28 | 0.69 | 1 | 0.64 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.25 | 0.98 | 1 | 0.88 | 0.2 | 0.08 | 0.09 | 0 | 0 | 0 | 0 | 0.31 | 0.99 | 1 | 0.98 | 0.25 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.25 | 0.89 | 1 | 0.91 | 0.89 | 0.91 | 0.74 | 0.74 | 0.74 | 0.74 | 0.91 | 1 | 1 | 0.43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0.44 | 0.95 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.98 | 0.54 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.16 | 0.21 | 0.36 | 0.84 | 0.83 | 0.6 | 0.6 | 0.5 | 0.19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2.** Raw Image (Left) and normalized image (Right)

### 2.3.2 Noise Reduction

After Normalization, we used the median filter to remove noise this is a nonlinear digital filtering technique to improve the image by removing especially Gaussian noise. We used Median Filter because it preserves the edge while removing the noise as edge is an important aspect of an image. Figure 3 shows the processed image of the median filter. An unwanted horizontal line connected to number “Zero” is removed after Median Filter.

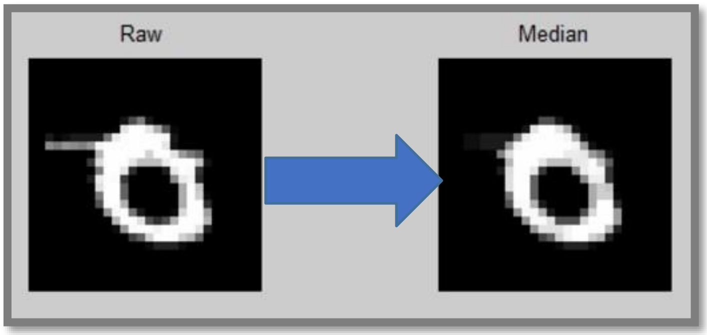


**Figure 3.** Raw Image (Left) and image after Median Filter (Right)

### 2.3.3 Image Sharpening

The next step in preprocessing is the image sharpening technique which uses a blurred, or "unsharp", negative image to create a mask of the original image. The unshaped mask is then combined with the positive (original) image, creating an image that is sharper than the original. Sharpening uses a filter that amplifies the high-frequency components of a signal. It’s a necessary step taken after Median Filter as Median Filter not only remove noise, but also weaken the entire image in general. Sharpening can restore or enhance some of the useful information weakened by Median Filter. Figure 4 shows the image sharpening after processing Median Filter.

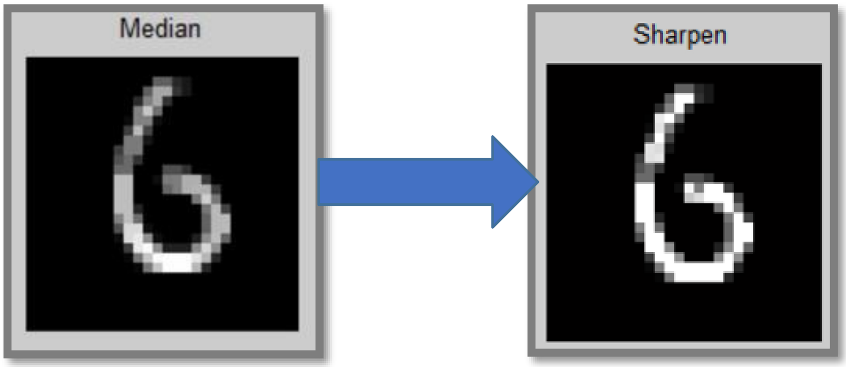


**Figure 4.** Median Filter and sharpened image

### 2.3.4 Image Attribute Reduction

Attribute reduction techniques is done to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.  Each original image has a total of 784 attributes. It would be beneficial to reduce the total attributes to a relatively small amount so that it’s more data efficient and easier to be processed. A direct way to reduce the attribute is by dividing the image into each block and finding the mean of that block as one attribute. Each block can be treated as one attribute.
The number of blocks determines the number of attributes. This is done for 2 x 2, 4 x 4, 7 x 7, and 14 x14 attributes (blocks) of the image. Accuracy is improved as the no of blocked are increased (proved later in this report).

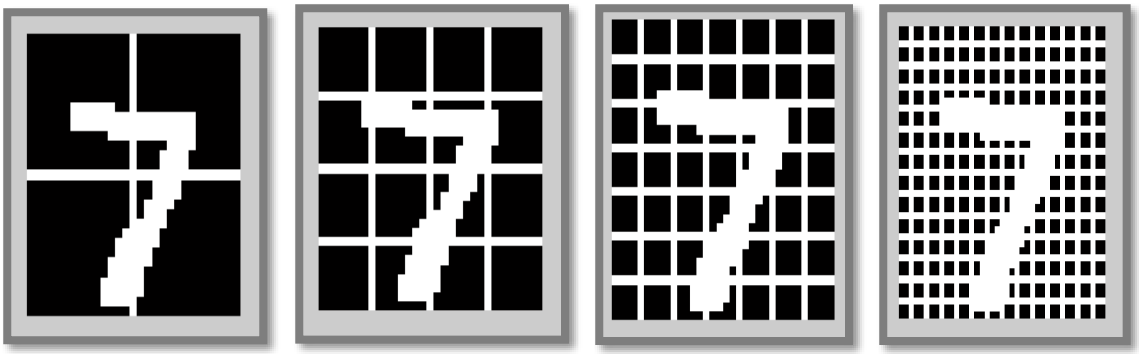


**Figure 5.** Image Attribute Reduction 2 x 2, 4 x 4, 7 x 7, and 14 x14

Figure 5 shows a visual representation on how the Image attribute reduction is done on the handwritten image. First the image is divided into equal blocks in a 2 x 2 block. Image attribute is reduced to 4 by taking mean for each block. A dramatic attribute reduction is achieved. Then, the same process is repeated for all 10,000 images. Apart from that, a 4x4, 7x7 and 14x14 attribute reduction is performed separately to compare and find the optimal number of attributes that best represent the image.

### 2.3.5 Add A New Attribute

We used BWEULER function in the MATLAB to extract a feature of the handwritten image, it returns the Euler number for the binary image. The Euler number is the total number of objects in the image minus the total number of holes in those objects. Objects are connected sets of on pixels, that is, pixels having a value of 1. Figure 6 is an example that BWEULER evaluates an image and provides a Euler number. No. of objects is 1 and the No. of hole is 1 so Euler No is 0.

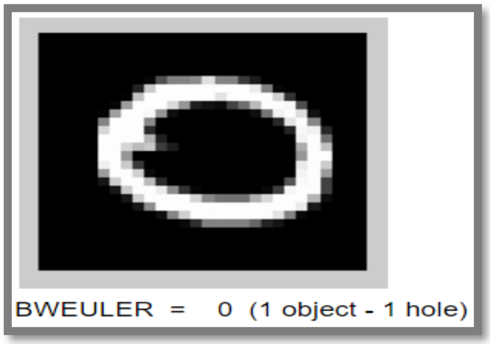


**Figure 6.** Sample Image and BWEULER Output for One Image

Figure 7 represents the data is run on 10,000 datasets. The portion of image “1” is clearly recognized as “1” than other portions of Number. From this, this new attribute could be added to the whole dataset and used to separate image “1” from other images

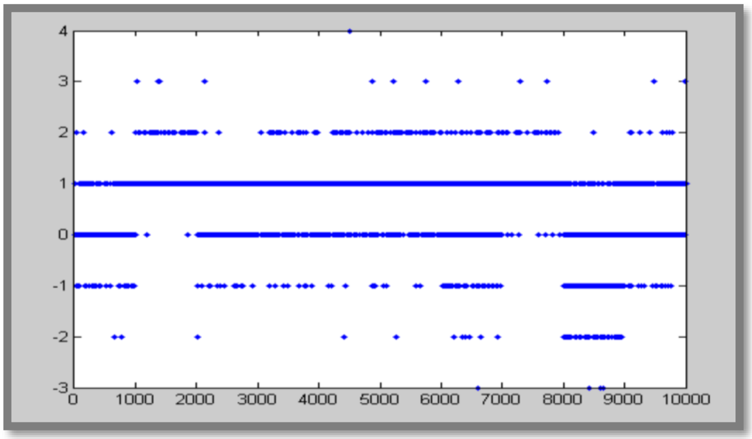


**Figure 7**. BWEULER output for Entire 10,000 Images

## 3. Results and Discussion Approach

In this section, the results of recognition system for offline handwritten Gurmukhi characters are presented. The K-Nearest-Neighbor has used in this study. The results are based on two feature extraction techniques, namely, diagonal and transitions features. As sated earlier, we have also experimented some partitioning strategies. The training and testing the data are done in Rapid Miner with 70% of the data in training and 30% for testing. The operator is first trained using the Training dataset and later the images in the testing dataset are send to the apply model operator to test the accuracy of the classifier.

Figure 8 shows the main process layout of the Rapid Miner. All the dataset (training and testing) is combined and split into 70:30 ratio get independent results before sending it to be training and testing. Various classification models were used to check the accuracy of the model like Neural Net, K-NN, Random Forest, Decision Tree and Bagging with gradient boost.

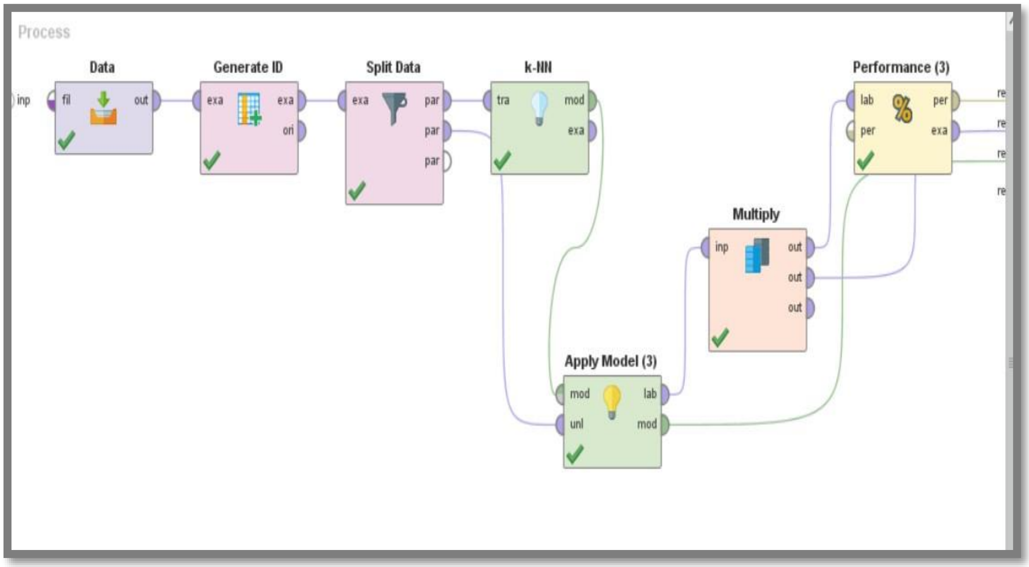


**Figure 8.** Main Process

### 3.1 Accuracy by Methods

To answer the first hypothesis, the same dataset is used with 70:30 train/test ratio and classified by various classification models.

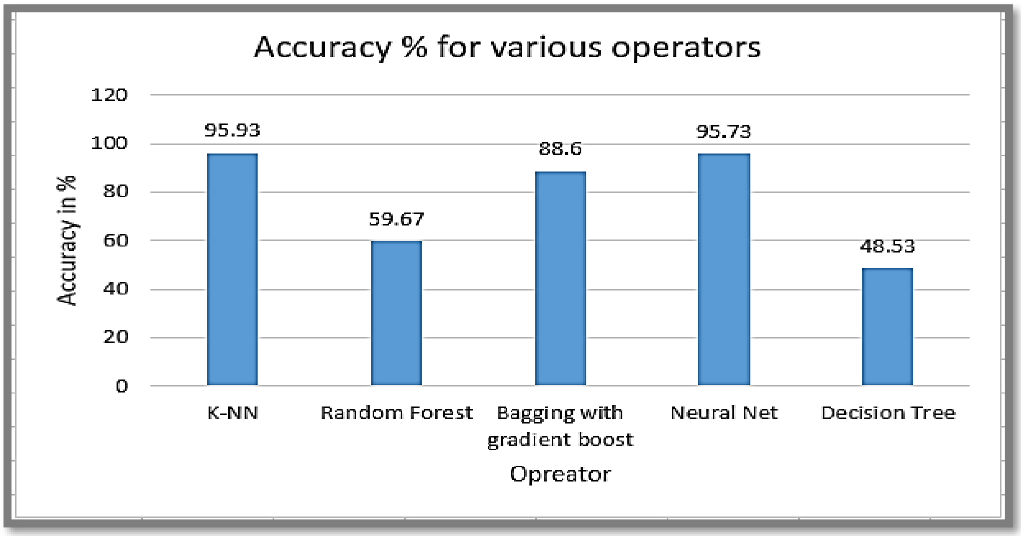


**Figure 9**. Accuracy in % for various operators

Figure 9 clearly shows that maximum accuracy is obtained by K-NN and Neural Net classifiers. Since, K-NN is significantly faster than Neural Net and its performance is similar as Neural Net, so we decided to use K-NN for prediction.

### 3.2 Accuracy by Attributes

Using the dataset obtained by Image Attribute Reduction in MATLAB (discussed earlier in the preprocessing section) analysis is done to check the accuracy of the classifier K-NN and Neural Net. The results in figure 10 were obtained with 70:30 train/test ratio by varying attributes length to 4, 16, 49 and 196.

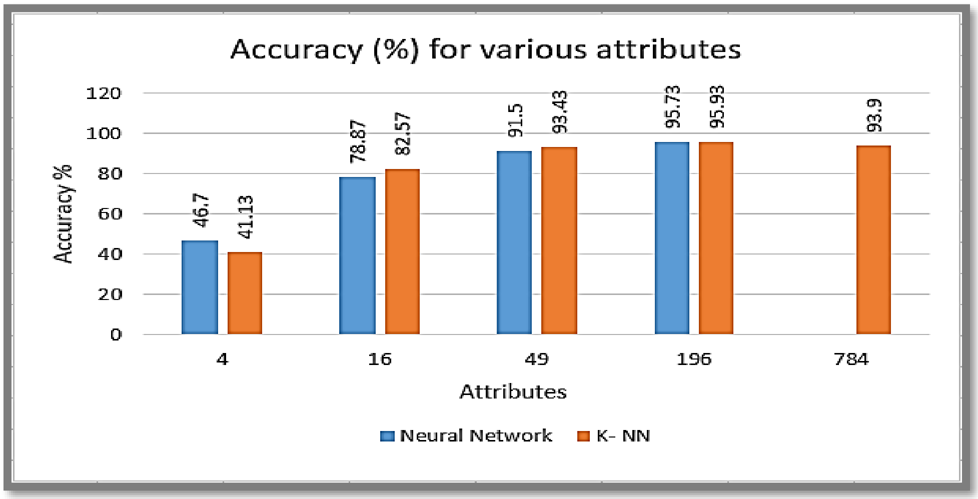


**Figure 10.** Accuracy in % for various Attributes

The graph clearly shows that when the attributes are increased the accuracy is also improved. With 196 attributes, we got an accuracy of 95.73% and 95.93% in Neural Net and K-NN respectively. It also shows that 784 attributes were reducing the accuracy compared to 196 attributes. It implies that more attributes preserve more information about the image and that helps classification, but excessive information carried by 784 attributes for instance did not improve the classification. Thus, we decided to use 196 attributes for both training and testing as it was the most efficient and sufficient representation of the image.

### 3.3 Accuracy by Training Ratio

To check the impact of the training data on the accuracy, we decided to change the ratio of the training and testing data using the reduced 196 attributes.

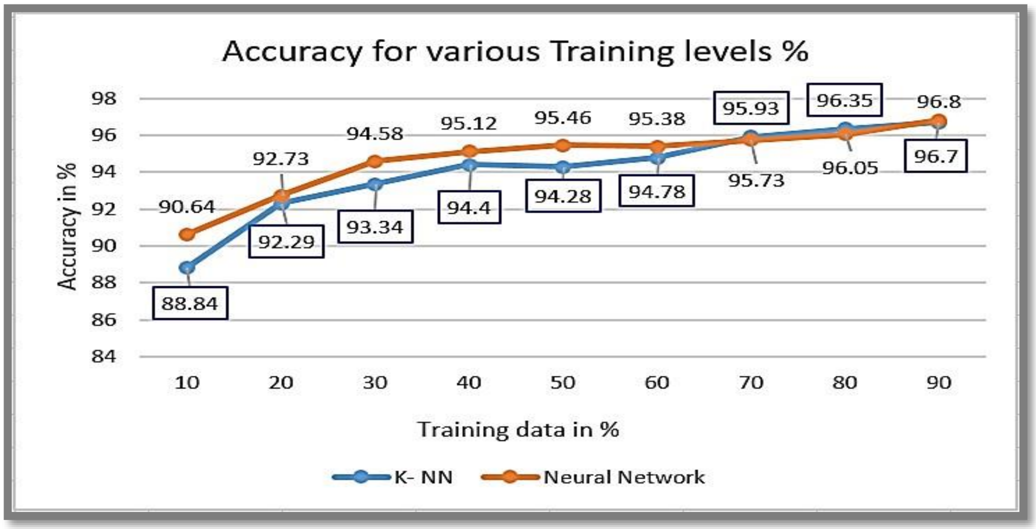


**Figure 11.** Accuracy for Various Training Levels in %

On performing the analysis in RapidMiner using 196 attributes for various training and testing ratios with classifier Neural Net and K-NN. Figure 11 clearly indicates that as the training data is increases the accuracy of the classifier is also improved. With 90:10 ratio of training and testing, a maximum accuracy of 96.8% and 96.7% were achieved by Neural Net and K-NN respectively. More data for training helps classification significantly.

#### 3.3.1 Difficulties

Some of the difficulties faced during the preprocessing were discussed in this section. The figure 12 above on the left shows the handwritten image which is not even recognized by us. The images in the middle and the right were handwritten digit one and zero, when we preprocess these images in Bewilder one is recognized as 2 (2 (objects) - 0 (holes) = 2) and zero is recognized as one (1 (object) - 0 (hole) = 1).

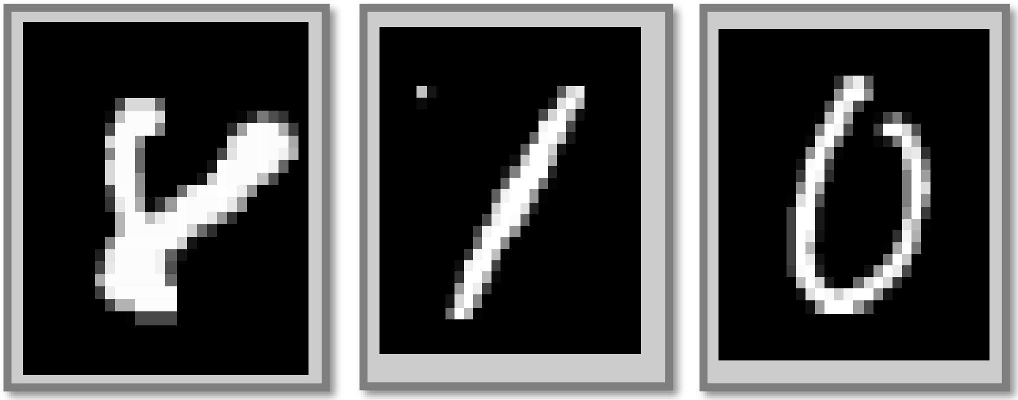


**Figure 12.** Handwritten Images (From Left) - Unknown No. 8, No. 1, and Zero

In these cases, bewilder is not an effective tool to preprocess the handwritten image. Because of the errors in the raw dataset the accuracy of the classifier is affected.

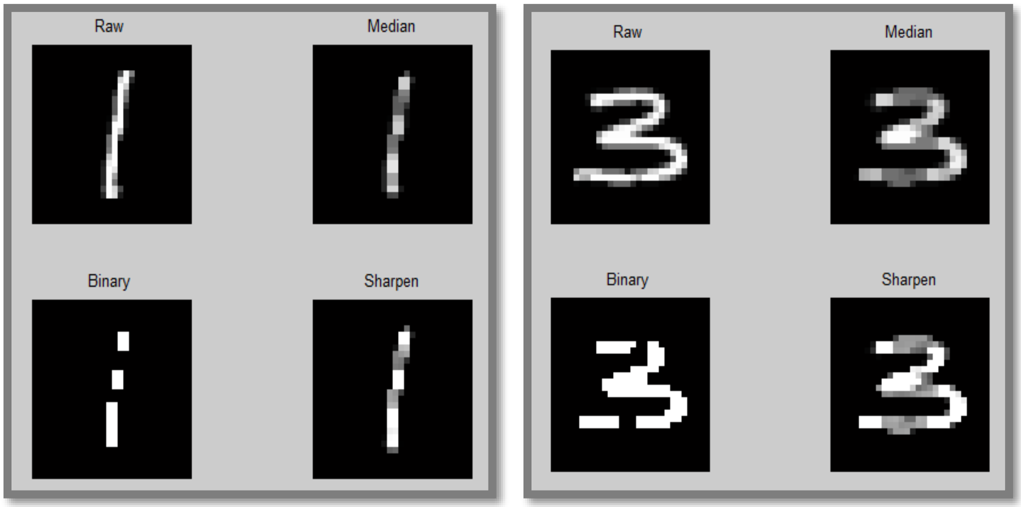


**Figure 13.** Problem in Noise Removal stage

Another problem which we faced in this project is during the preprocessing stage. Figure 13 shows the Handwritten Image One and Three. In there, the median filter blurs the image. Sharpening helps restoring some details back after median filter, but the image is still blurred in an extent as compared to the raw image which exacerbates the quality of good image.

Another observed problem is the use of Binary technique. Binary technique transforms all image pixels to either 0 or 1. It simplifies the data and offers clean looking of the data if it is controlled well. However, from Figure 13 we can see that the good image (one and three) are in fact damaged by doing Binary process. That's because the binary threshold was not set correctly, and image loss was resulted in. As it's difficult to set a proper threshold for all images, we decided not to use Binary technique and had less damage to the images.

### 3.3.2 Application.

A real-time testing was performed by our team members to test the accuracy and validate the effectiveness of the classification and preprocessing method.

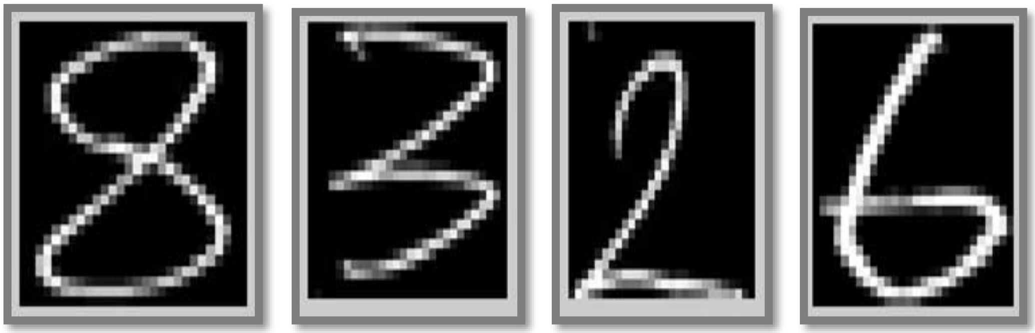
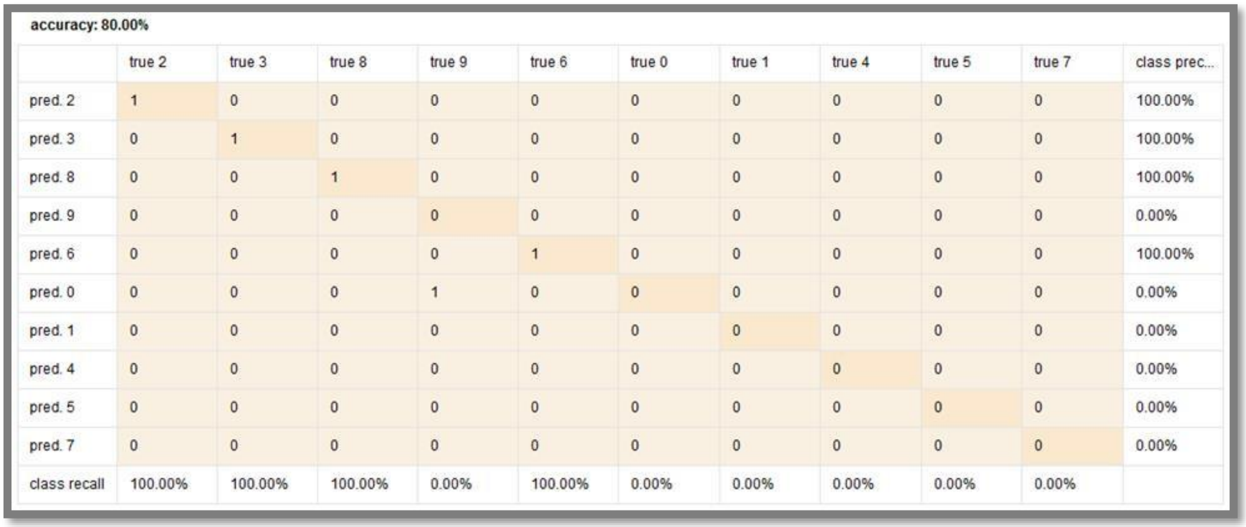


**Figure 14.** Handwritten digits

Figure 14 shows some of the handwritten digits written by us and we used that for testing in the RapidMiner. Each subject was required to write a single number on a blank paper. A photo of that number was taken by a smartphone camera to obtain the digital image. Then It was resized to 28 x 28 pixels and converted it to Black/White before sending it to be testing in RapidMiner as in Figure 14.



accuracy: 80.00%

| | true 2 | true 3 | true 8 | true 9 | true 6 | true 0 | true 1 | true 4 | true 5 | true 7 | class prec... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| pred. 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. 8 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 6 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| pred. 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| class recall | 100.00% | 100.00% | 100.00% | 0.00% | 100.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | |

**Figure 15.** Handwritten digits

The overall accuracy of the handwritten image was 80%. Because it predicted Digit 9 as Digit 0. Figure 15 shows the confusion matrix of the output. The performance was acceptable considering the small sample size we took. Also, since the raw data from the MNIST database, we used for training had been already processed, if we do more processing on the testing image we wrote, the accuracy can be improved further.

## 4. Conclusion.

This paper has practiced different machine learning technique and different models for data training attempting to discover a representation of isolated handwritten digits that allow their effective recognition and to achieve the highest accuracy of predicting handwritten numeral. Thus, this study settled on classifying a given handwritten digit image as the required digit using five different algorithms and consequently testing its accuracy. This study built handwritten recognizers evaluated their performances on MNIST (Mixed National Institute of Standards and Technology) dataset and then improved the training speed and the recognition performance. In addition to develop a system for word based handwriting recognition system and test the handwriting of a given word and detect the writer by selecting which is being recognized for most of the user for a given training sample. This study discusses in detail all advances in the area of handwritten character recognition. The most accurate solution provided in this area directly or indirectly depends upon the quality as well as the nature of the material to be read. Various techniques have been described in this paper for character recognition in handwriting recognition system. The result of this study shows that accuracy is improved as the no of blocked are increased. Apart from that, a 4x4, 7x7 and 14x14 attribute reduction is performed separately to compare and find the optimal number of attributes that best represent the image. The initial hypothesis was answered with concrete data support. Comparing all the classification models we tested K-Nearest Neighbor is the preferred choice in terms of its high accuracy and computational efficiency. However, there is no single classifier that works best on all given problems. A result shows that probabilistic methods suit better for handwriting recognition. By varying the training and testing ratios (from 10% to 90%) we found that the larger training data size improves accuracy, but smaller testing dataset may also favor better accuracy. Preprocessing such as Attribute reduction (784 reduced to 196) reduce runtime and increase accuracy (from 93.2% to 95.9%). The proposed algorithm tries to address both the factors and well in terms of accuracy and time complexity. The general accuracy of tested K-NN was found to be 96.7% while 96.8% were achieved by Neural Network respectively. The overall highest accuracy 96.8% is achieved in the recognition process by Neural Network with the sacrifice of significantly extended runtime. This work is carried out as an initial attempt, and the aim of the paper is to facilitate for recognition of handwritten numeral without using any standard classification techniques. Image processing techniques Median filter, binary, Bweuler, and sharpening improve image quality, but we should be cautious to implement them as it may cause new problems in some cases. Handwritten digit recognition system can be extended to a recognition system that can also able to recognize handwritten character and handwritten symbols. Future studies might consider hardware implementation of the recognition system. In the future, we are planning to further explore the topic of data classification in perhaps other industrial related applications.

## Reference

A.U. Chukwu and K.A. Adepoju, On the power efficiency of artificial neural network (ANN) and the classical regression model, Progress in Applied Mathematics 3(2) (2012), 28–34

Ahmed, M., Rasool, A. G., Afzal, H., & Siddiqi, I. (2017). Improving handwriting-based gender classification using ensemble classifiers. Expert Systems with Applications, 85, 158-168.

Babu, U. R., Chintha, A. K., & Venkateswarlu, Y. (2014). Handwritten Digit Recognition Using Structural, Statistical Features and K-nearest Neighbor Classifier. International Journal of Information Engineering and Electronic Business, 6(1), 62.

Burel, G., Pottier, I., & Catros, J. Y. (1992, June). Recognition of handwritten digits by image processing and neural network. In Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 3, pp. 666-671). IEEE.

D. Gorgevik and D. Cakmakov, "Handwritten digit recognition by combining SVM classifiers," in Proceedings of the International Conference on Computer as a Tool (EUROCON '05), vol. 2, pp. 1393–1396, Belgrade, Serbia, November 2005. View at Publisher · View at Google Scholar

Ebrahimzadeh, R., & Jampour, M. (2014). Efficient handwritten digit recognition based on histogram of oriented gradients and svm. International Journal of Computer Applications, 104(9).

Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

Lawgali, A. (2016). Recognition of Handwritten Digits using Histogram of Oriented Gradients. International Journal of Advances Research in Science, Engineering and Technology, 3, 2359-2363.

Lee, Y. (1991). Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. Neural computation, 3(3), 440-449.

Liu, C. L., Nakashima, K., Sako, H., & Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition, 36(10), 2271-2285.

M. Paliwal and U.A. Kumar, Neural networks and statistical techniques: A review of applications, Expert Systems with Applications 36(1) (2009), 2–17.

O.D. Trier, A.K. Jain, & T. Taxt. (1995). Feature extraction methods for character recognition—a survey.

Salvador España -Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2014

Singh, P. K., Sarkar, R., & Nasipuri, M. (2016). A study of moment-based features on handwritten digit recognition. Applied Computational Intelligence and Soft Computing, 2016, 4.

Tuan Trung Nguyen, "Adaptive Classifier Construction: An Approach to Handwritten Digit Recognition", J.J. Alpigini et al. (Eds.): RSCTC, LNAI 2475, Springer-Verlag Berlin Heidelberg, pp. 578–585, 2002

U. R. Babu, Y. Venkateswarlu, and A. K. Chintha, "Handwritten digit recognition using K-nearest neighbor classifier," in Proceedings of the World Congress on Computing and Communication Technologies (WCCCT '14), pp. 60–65, Trichirappalli, India, March 2014. View at Publisher · View at Google Scholar · View at Scopus

Wang, J., & Yan, H. (June 2000). A hybrid method for unconstrained handwritten numeral recognition by combining structural and neural "gas" classifiers. Science Direct- Pattern Recognition Letters, 625-635.

## Biographies

**Shengfeng Chen** is a PhD student, in Industrial Engineering at Western Michigan University, Michigan, USA. He earned B.S. in Electrical Engineering from Hubei University of Technology, Wuhan, Masters in Electrical and Computer Engineering from Western Michigan University, Michigan. He is the member of Alpha Pi Mu and ETA KAPPA NU the honor society. His research interests include statistical process control, image simulation, machine vision system, artificial intelligence, and statistical quality control.

**Rabia Alamamlook** is PhD student industrial Engineering, and Master of Stats and Master of Science in Industrial Engineering. Mrs. AL-Mamlook earned B.S. in Mechanical Engineering from Tripoli University of Engineering, Masters in Stats from Western Michigan University and Engineering Management from Tripoli University. She has published journal and conference papers. Her research interests include Stats, simulation, , Quality, Machine learning and Engineering Management. Mrs. AL-Mamlook is member of ASQM, ASEM.

**Yuwen Gu** is a PhD Candidate of Industrial Engineering at Western Michigan University. She holds a M.S. degree of Industrial Engineering from Western Michigan University, and a B.S. degree from East China Normal University, Shanghai. Her dissertation aims to evaluate and improve the public health system in the state of Michigan. Her research fields of interest include decision-making in healthcare management system and education system. She has a strong background of simulation modeling, data science and operation research.

**Dr. Lee Wells** is an Assistant Professor, in Department of Industrial and Entrepreneurial Engineering and Engineering Management at Western Michigan University. Professor Wells received his B.S. and MS degrees in Mechanical Engineering at Michigan Tech and PhD in Industrial and Systems Engineering at Virginia Tech. His research interests include: advanced visualization techniques for quality assessment, quality monitoring techniques for high- dimensional data, and active monitoring and control for multistage assembly systems.