Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

Handwriting digit recognition using Decision Tree Classifiers

Demir Korać[1], Samed Jukić[1], Mujo Hadžimehanović[1]
[1]International Burch University, Sarajevo, Bosnia and Herzegovina
demir.korac@stu.ibu.edu.ba
samed.jukic@ibu.edu.ba
mujo.hadzimehanovic@stu.ibu.edu.ba

***Abstract* – The usage of handwritten character recognition has been useful for usage from large to common consumer usage. The transitional period of the handwritten to the digital age can be largely improved by focusing on perfecting handwritten character recognition. This paper and work aims to focus on handwritten digit recognition using the decision tree classifier machine learning method, implemented, trained and tested on the data set gathered from the Modified National Institute of Standards and Technology dataset. The data to be recognized is inputted from a pre-existing reliable set, used both for training and testing, in order to give a fair result. The system is run through a Python script and the data set is stored in CSV format, preprocessed and ready for further usage. Taking into consideration the size of the dataset (42000 rows of data), the system's overall performance is satisfactory with an accuracy of 85% and outputs the results in an understandable manner.**

Keywords – character, decision, handwritten, recognition

1. Introduction

Optical character recognition is the process of mechanically or electronically converting typed, printed or handwritten images of text into a machine-encoded text. The source can be taken from a photo of a document, a scanned document or a scene photo (billboards in a landscape photo). Another possible source is superimposed text on an image, such as subtitle text from a television broadcast. This process of recognition and the technique used in it is called a handwriting recognition system. In literature this recognition is classified into offline handwriting recognition and online handwriting recognition depending on the style of recognition. If an image of handwriting is previously acquired and recognized after it will be classified as offline recognition. However if the handwriting is inputted directly into the machine to be recognized this is called online recognition. One way of doing this is writing on a touchpad or other device dedicated for writing and having it recognized. Another classification exists regarding the recognition and it is based on the process of recognition itself and the technique.

We can recognize two main categories: segmentation free and segmentation based recognition. Segmentation free recognition is based on recognizing the character without segmentation into smaller units or characters,

i.e. words into characters. Segmentation based recognition is the opposite of this. In this approach each word is segmented into smaller units/characters and each character is recognized separately.

Handwriting has been for the most part of our history, the primary means of communication and information organization, but with the modernization of these fields handwriting is becoming slowly obsolete. However the legacy of handwritten information cannot be overlooked, so a transitional period and process needs to occur.

That's where handwriting and optical character recognition comes into play. The importance of handwriting recognition is reflected in many industries such as:

health care (7,000 people are killed per year by the poor handwriting of doctors)

automotive (digital handwriting solutions allow drivers to write characters or numerals, or simply gesture with their fingertips on vehicles' onboard computer screens instead of typing on a standard keyboard)

field services (field service technicians use HWR technology for digital data capture, decreasing paperwork and information loss while also allowing technicians to see precise notes on customer history)

education (using HWR technology, students can benefit from more than just the increased comprehension linked to taking handwritten notes. For example, handwriting recognition tech can take your sloppy algebra equation, convert it into neat, digital text, and then crunch the numbers in a matter of seconds.)

consumer (with the rapid success of tablets and smartphones, the market is desperately in need of an alternative to inaccurate, digital keyboards and HWR tech is the best remedy.)

2. Previous work

Perwej Y. and Chaturvedi A. have worked on recognizing the English alphabet handwriting using Artificial Neural Networks using binary pixels of the alphabet to train the Neural Network and the accuracy of this method is 82.5% [1]. The data set that they worked on are handwritten English alphabet characters which are scanned from documents and then "cleaned" and "smoothed". The characters are then split into 25 segment grids, scaling and thinning the segments of the characters to obtain skeletal patterns. This is then transformed into binary values representing the segments input. (Shown in Figure 1.)

Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

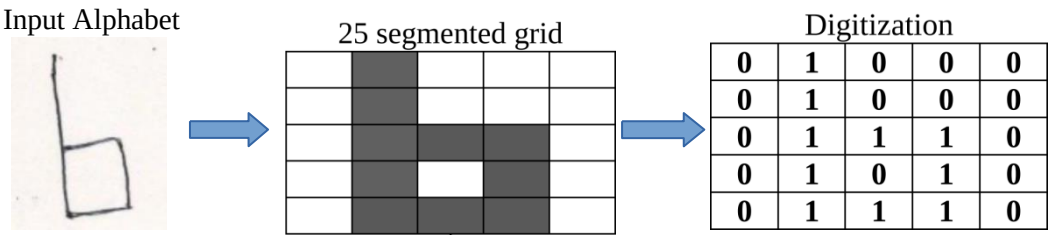

Figure 1. Process of recognition

Amrouch M. and Y. Es saady have worked on a method using sliding window based on the Hough transform as a feature extraction technique [2]. An inputted image is divided into two windows and then a dominant direction is determined based on the Hough transform and a directional feature vector sequence has been formed. This method is based on continuous HMMs and directional features with an average accuracy of 90.4%.The data set that they worked on is a database of Amazigh printed characters, containing 240 isolated characters. The images of characters are in Gray level of dimensions 96x96 pixels.

M. Hanmandlu and O.V. Ramana Murthy [3] have done a study on recognition of Hindi and English numerals. The numerals are represented in the form of exponential membership functions which serve as a fuzzy model. They have achieved an overall recognition rate of 95% for Hindi numerals and 98.4% for English numerals. The data set that they worked on is a database of 5000 samples of numerals for handwritten English numerals. For Hindi they used a database of totally unconstrained handwritten numerals created using the services of a large number of writers, since there is no standard database available at the moment for handwritten Hindi numerals.

Nafiz Arica at al. has proposed a method of recognition without pre-processing which he believed leads to loss of necessary information [4]. This has been backed up with a powerful segmentation algorithm with utilization of character boundaries, maxima and minima, slant angle, upper and lower baselines, stroke height and width and ascenders and descenders which improved the search algorithm of the optimal segmentation path, applied on a gray-scale image. The dataset used was the handwritten database of Lancester-Oslo/Bergen, which contains single author cursive handwriting. In this dataset, 1,000 words with lower-case letters are segmented and used for HMM training and another disjoint set of 2,000 words are used for testing performance of the proposed system.

Table 1. Results of recognition for the LOB Dataset

	TEST DATA SIZE	**LEXICON SIZE**			
		50	1000	30000	40000
LOB Dataset	2000	92.3	90.8	89.1	88.8

The overall recognition rate of the whole system on word basis for various lexicon sizes is shown in the table above.

Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

3. Method and Materials

The dataset being used in the project is the famous Modified National Institute of Standards and Technology database of handwritten digits [5], as it is the most reliable and largest database of this type. This dataset is taken as a subset of another larger dataset by NIST. The MNIST database of handwritten digits consists of 42000 rows of data. It will be split into a 80/20 ratio for training and test data respectively, so the training data will have 33600 rows and test data will have 8400 rows of data. The digits have been size-normalized and centered in a fixed-size image. The database and the files such as the training set images and labels, and the test set images and labels can be found at: http://yann.lecun.com/exdb/mnist/

The machine learning model used will be a Decision Tree Classifier implemented through the scikit-learn machine learning library in Python [6]. Decision Trees are used in data mining, machine learning and statistics as a predictive modeling approach. The structure of trees is as follows: Class labels are represented as leaves and the conjunctions of features that lead the class labels are represented as branches. The process of operation will be such that the dataset will be inputted as a matrix containing cell inputs from the database as intensity values of pixels of 28x28 images.

Table 2. Sample from dataset

label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9
1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0

An empty classifier will be created and using the fit method it will be filled with the training data. After the classifier finishes with the training data, we will move on to the rest of the data set, the testing part. Using the predict method we will output the classifier prediction of the handwritten digit from the test dataset along with the actual image of the digit.

Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

4. Process

The process of recognition is as follows:

The dataset is imported and separated into two parts with a ratio of 80/20. The train set consists of 33600 and the test set consists of 8400 rows of data.

Splitting the dataset:

```
traindata=data[1:33600,1:]
train_label=data[1:33600,0]
```

The data and its label are fitted to the model to learn from it. We do this by using the fit method and passing our training set to it.

Fitting the training set:

```
clf.fit(traindata,train_label)
```

The user of the system gives an input of the order number he wants to test the system with, from the dataset. That number is taken and a sample with that order number is chosen from the dataset, along with its actual label.

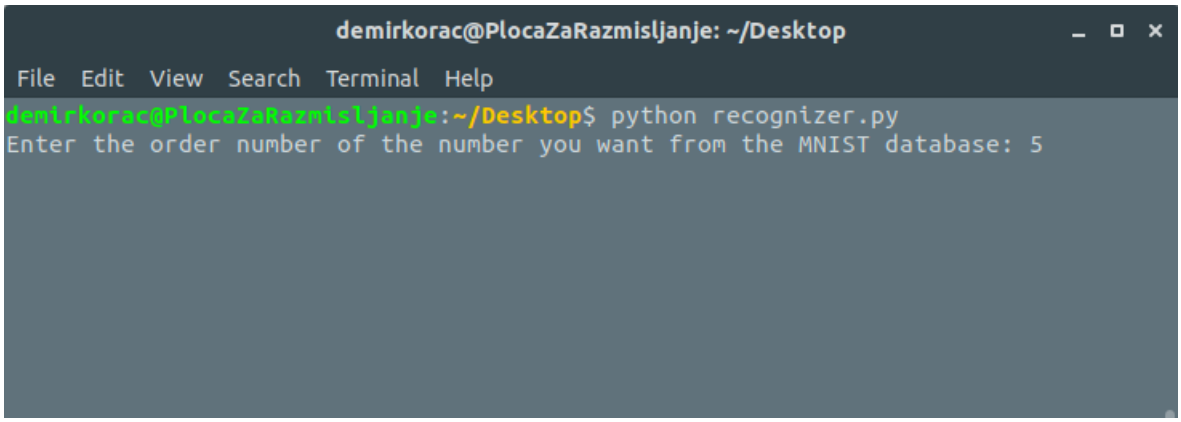

Figure 2. Taking input from the user

Using the predict method the sample is predicted from the sample and the prediction is outputted. Using the shape method we are taking the sample from a row vector and shaping it as a 28x28 matrix of pixel intensity values. We are then creating a figure, which will be displayed after prediction.

Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

Using the shape method:

```
d.shape=(28,28)
```

When the figure is created, we are proceeding with the prediction. We will then be using the predict method from the Decision Tree Classifier in the scikit-learn library [7].

Using the predict method along with output of the data:

```
print ("The predicted digit is =", clf.predict( [testdata[number]]))
print ("The actual digit is =", actnum)
```

Along with predicting the digit, we will also be outputting the actual label of the sample number that the user chose. An extra feature added is the current certainty of prediction calculation.

Certainty of prediction:

```
p=clf.predict(testdata)
count=0
for i in range (0,8400):
    count+=1 if p[i]==actual_label[i] else 0
print ("Certainty of prediction=", (count/8400)*100,"%")
```

For this feature we are taking all numbers in the range of the test data set and using the predict method, as to calculate the success rate of our model on the current test data.

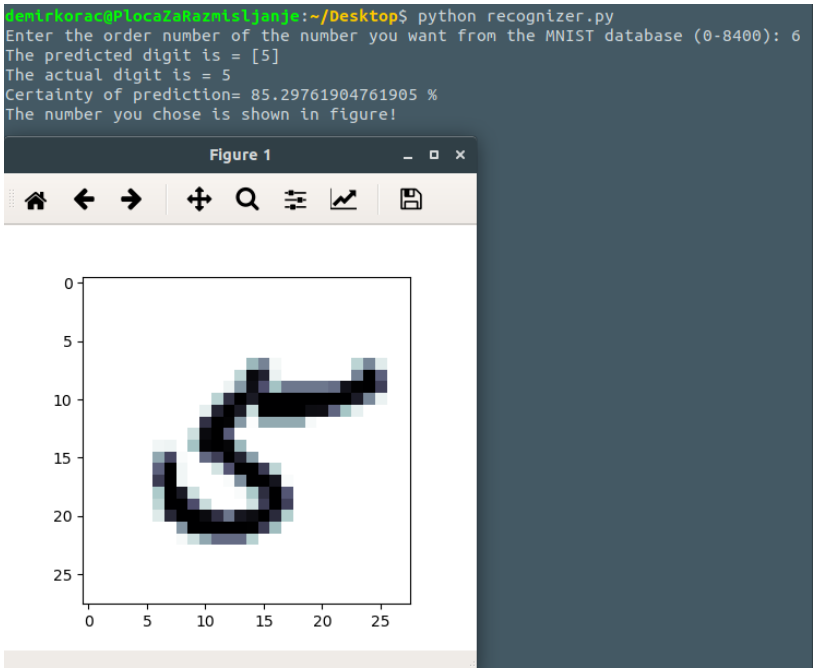

Figure 3. The systems' process from the users' perspective

Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

5. Usage workflow

In Figure 4. We can see the workflow according to which the system will be operating.

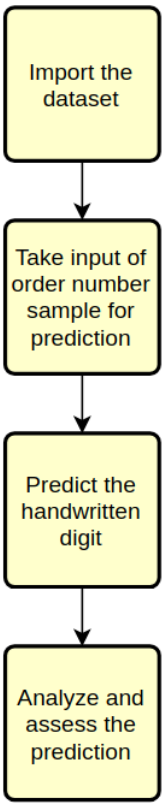

Figure 4. Usage workflow

Since the system does not need dataset input from the user, we will just be importing the dataset. The system needs user input, since the user will be choosing the digit that is to be recognized, so he needs to provide the order number from the data. The system then outputs its prediction along with other data such as the actual label and the prediction certainty. This allows the user to analyze the data given to him, and check if the prediction was correct and is the prediction certainty satisfactory.

4. Decision Tree Classifier in scikit-learn

A decision tree is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in a recursive manner called recursive partitioning. This flowchart-like structure helps you in decision making. It's visualized like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

Journal of Natural Sciences and Engineering, Vol. 2, (2020)
DOI number: 10.14706/JONSAE2020214

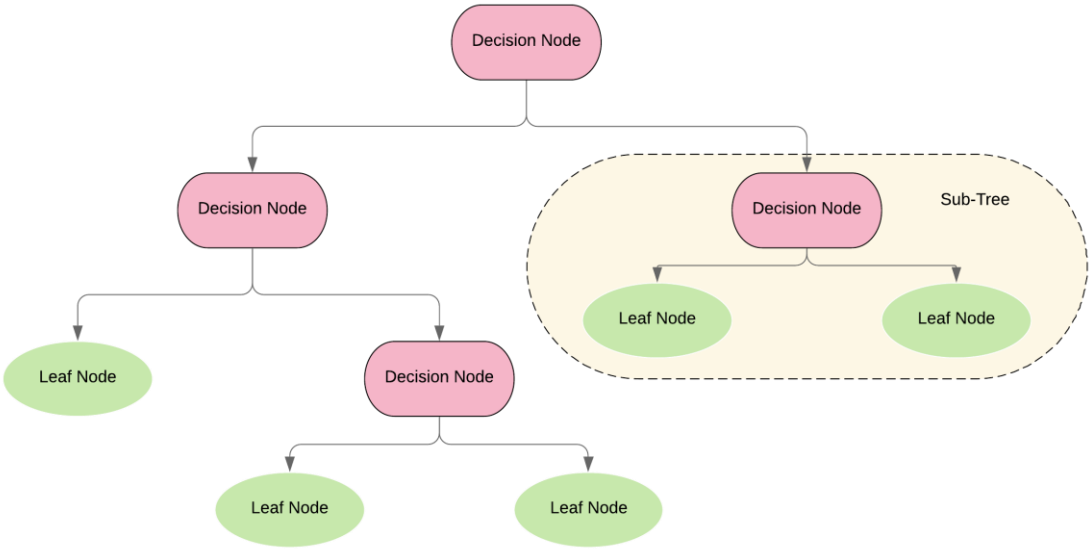

Figure 5. Decision tree operation diagram

5. Results

After processing and thorough testing the model seems to have a success rate of ~85% on the test dataset given to it. The system itself may have some confusion when analyzing similar digits. This occurs because the digits themselves are handwritten and because the data has been gathered from over 250 different writers, and some of them write different digits in a similar manner. An example is shown in Figure 6. where we can see that the system is having trouble predicting number 9, but instead predicts 3. From the image we can see that the digits themselves are quite similar.

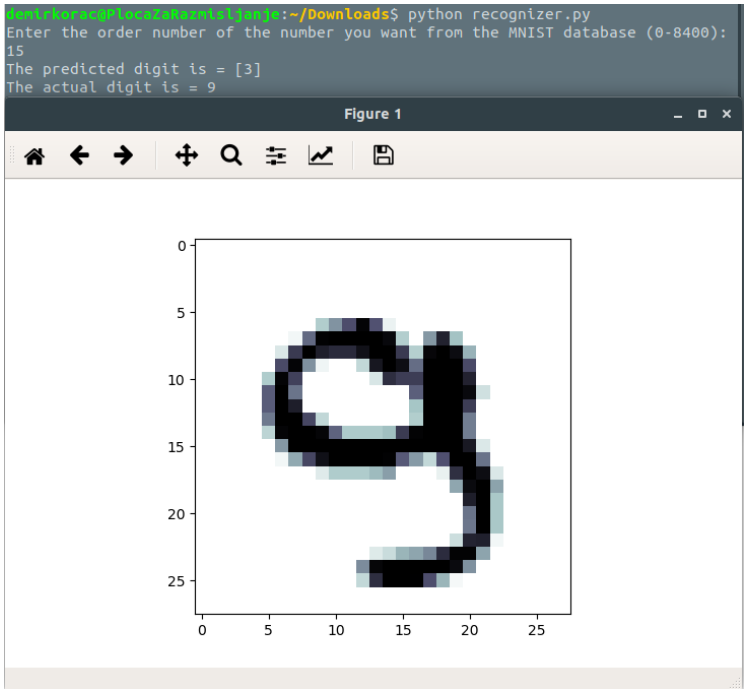

Figure 6. Wrong prediction of digit 9 as digit 3

Different parts of the world write digits differently, so the algorithm would be much more accurate if it would be implemented based on location, and also, if different Machine Learning methods or Ensemble methods would be used, we would surely improve accuracy. Finally the thing that would take the recognition to the next level is image quality, considering that this algorithm has been implemented on a dataset with quite low quality of images.

5. Conclusion

It should be noted that this data set is not sufficient for a higher level model, to be used in production, but as its using a small dataset, used for initial training, it has shown to be a pretty successful model and method.
If we would train the aforementioned algorithm on a larger dataset with higher image quality, it could be used for real world projects. The confusion of the system may also be affected by the quality of the images in the given dataset, as it is limited to only 28x28 matrices. Future training with a dataset with matrices of larger magnitude may prove more successful.

REFERENCES

[1] Perwej Y. & Chaturvedi A. -Neural Networks for Handwritten English Alphabet Recognition. 2011 IJCA

[2] M. Amrouch, A. Rachidi, M. El Yassa, D. Mammass - Printed amazigh character recognition by a hybrid approach based on Hidden Markov Models and the Hough transform. 2009 IEEE https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[3] M. Hanmandlu, O.V. Ramana Murthy - Fuzzy model based recognition of handwritten numerals. 2007 Pattern Recognition

[4] N. Arica, F.T. Yarman-Vural - Optical character recognition for cursive handwriting. 2002 IEEE

[5] MNIST handwritten digit database - http://yann.lecun.com/exdb/mnist/

[6] Decision Tree Classification in Python - https://www.datacamp.com/community/tutorials/decision-tree-classification-python

[7] sklearn.tree.DecisionTreeClassifier-