Arabic Digit Feature Extraction And Classification

Parnian Taheri[a]

[a]*Sharif University of Technology, , Tehran, , ,*

Abstract

This study investigates the efficacy of various classification and feature extraction techniques for the recognition of handwritten Arabic digits using the Hoda dataset. The dataset comprises a diverse collection of handwritten digits, presenting a challenging scenario for digit recognition tasks. The research explores multiple classification algorithms, including k-Nearest Neighbors (k-NN), Decision Trees, and others. Additionally, various feature extraction methods such as Zoning and Histogram are evaluated for their impact on classification accuracy. Moreover, to check the robustness of the model the noise is added to the testset. The effects of this preprocessing steps on the classification results are thoroughly examined.

1. Introduction

Handwritten digit recognition has been a longstanding challenge in the realm of pattern recognition and machine learning, finding applications in various fields, from optical character recognition to automated postal sorting. While extensive research has been conducted on Latin digit recognition, the exploration of Arabic handwritten digits presents a distinctive set of challenges. The complexities arise from the fluid nature of Arabic digit handwriting, introducing variations in shape and structure. Bridging this gap is crucial to address the diverse linguistic and cultural contexts where Arabic digits are prevalent. Recognizing handwritten Arabic digits is particularly complex due to the diversity in writing styles and inherent variations in shape and structure. In contrast to the established body of research on Latin digits, there exists a notable gap in the literature concerning the recognition of Arabic digits.

This research focuses on the recognition of handwritten Arabic digits, utilizing the Hoda dataset, which encompasses a rich collection of such digits. We delve into the evaluation of various classification algorithms, including k-Nearest Neighbors (k-NN), Naive Bayes, and others, with a specific emphasis on their effectiveness in recognizing Arabic digits.

Additionally, we explore feature extraction techniques such as zoning and histogram analysis, aiming to capture unique characteristics inherent in Arabic digit writing styles. To enhance practical applicability, we introduce preprocessing techniques, including the addition of noise, to evaluate the robustness of our recognition models.

2. Hoda dataset

Hoda dataset is the first dataset of handwritten Farsi digits that has been developed during an MSc. project in Tarbiat Modarres University entitled: Recognizing Farsi Digits and Characters in SANJESH Registration Forms. This project has been carried out in cooperation with Hoda System Corporation. It was finished in summer 2005 under supervision of Prof. Ehsanollah Kabir. Samples of the dataset are handwritten characters extracted from about 12000 registration forms of university entrance examination in Iran.

The dataset specifications is as follows:

Resolution of samples: 200 dpi
Total samples: 102,352 samples
Training samples: 60,000 samples
Test samples: 20,000 samples
Remaining samples: 22,352 samples

Number of samples per each class:
0: 10070
1: 10330
2: 9923
3: 10334
4: 10333
5: 10110
6: 10254
7: 10363
8: 10264
9: 10371

3. Zero Padding

The Hoda Handwritten Digit Dataset, like many datasets, presents a challenge in terms of varying image sizes. Handwritten digits may be written with different scales and proportions, impacting the consistency required for effective model training. To address this, a pre-processing step is introduced to standardize the images to a uniform size.

In this pre-processing step, images are resized to a common dimension by adding a white border around them. This white border serves as a padding, ensuring that all images share the same dimensions without distorting the original content. The size of the white border is determined in a way that it ensures

that both the height and width of each image are extended to match the maximum dimension found in the dataset. Finally, the size of the image becomes 54 x 45.

4. Brief description of the used classification techniques

In this section, a brief description of each used classification techniques is presented. In the results section, the accuracy of each classifier/features combination is reported as well. Some of the used classification techniques have parameters that need to be specified (e.g. k in k-nn classifier). Such parameters are optimized using a validation set. The validation set is composed of 1,000 samples of the total 4000 samples that we took from Hoda database.

4.1. K-Nearest neighbor

The K -Nearest Neighbor (KNN) [14] is one of the simplest classification techniques; yet gives surprisingly high accuracy. In KNN, there is no training stage. All training samples must be present in the testing phase; and Euclidean distances between each training sample and the sample to be tested are calculated. The K training samples that have the smallest distances to the test sample are found and their classes are identified. To choose the best k we classify the data for different 'k's and test with the validation set and choose the one that gives the highest accuracy.

4.2. Bayes

The Bayes Classifier, rooted in Bayesian probability theory, is a powerful statistical method for classification tasks. At its core, the Bayes Classifier assigns a class label to an input based on the likelihood of that input belonging to each class. The decision is made by considering both prior probabilities and the likelihood of observed features given each class. In the context of handwritten digit recognition, the features could be pixel values or higher-level descriptors extracted from the digit images.

4.3. Decision Tree

The Decision Tree Classifier is a versatile and interpretable machine learning algorithm widely used for both classification and regression tasks. Its appeal lies in its ability to create a structured decision-making process, resembling a tree-like flowchart, making it intuitively understandable. A Decision Tree is composed of nodes that represent decision points and leaves that correspond to class labels or regression values. Each internal node tests a specific feature, and the branches emanating from the node represent the possible outcomes of the test. The final predictions are made by traversing the tree from the root to a leaf node based on the features of the input data. While Decision Trees offer interpretability and flexibility, they may be prone to overfitting, capturing noise in the training data. This challenge can be mitigated through techniques such as pruning or by using ensemble methods.

4.4. Random Forest

The Random Forest Classifier is a powerful ensemble learning method that leverages the strength of multiple Decision Trees to improve predictive accuracy and robustness. It addresses the limitations of individual trees, such as overfitting, by combining their outputs through a process called bagging (Bootstrap Aggregating). In addition to sampling data, each tree considers a random subset of features at each split. This feature randomization further promotes diversity among the trees. While Random Forests are robust, they may be computationally expensive, especially with a large number of trees.

5. Brief description of the used feature extraction techniques

5.1. Zoning

In this technique, the image is uniformly partitioned into 9×9 zones [23]. The average of each zone is calculated leading to a feature vector of 81 elements. See Fig. 1.

5.2. Vertical and Horizontal Histogram

In this technique, the horizontal and vertical histograms of the image are calculated leading to a feature vector of 99 elements. See Fig. 2.

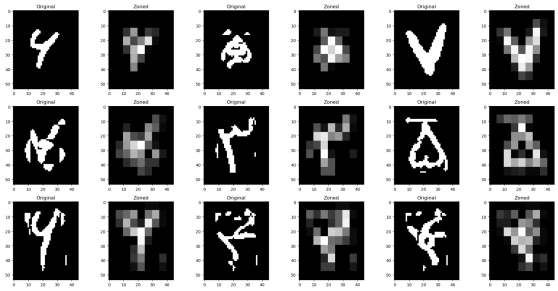

Figure 1: Zoning

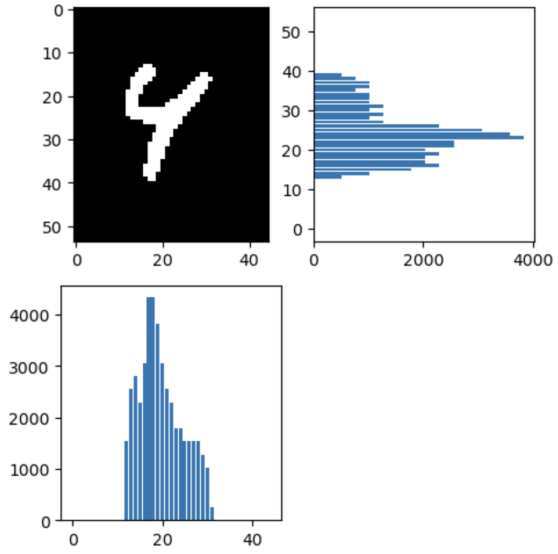

Figure 2: Histogram

6. Add Noise

In this part, in order to test the robustness of the model the salt and pepper noise is added to the validation set. Firstly, we find the best k for k-NN classification. After that we classify the validation set and check the accuracy.

7. Find k

To set the value of k in KNN classifier, we do classification for some 'k's and save the k, which gives the best performance. See Fig. 3 and 4.

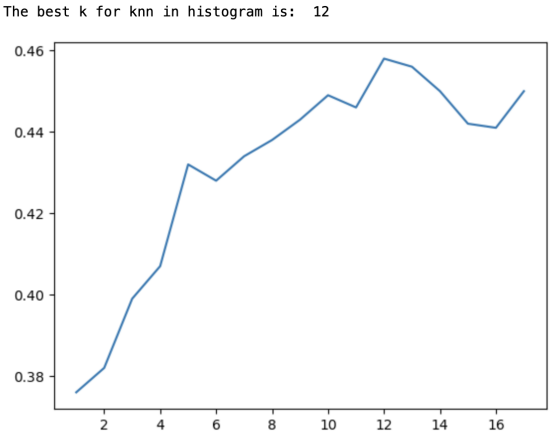

Figure 3: The diagram to find the best k in histogram feature extraction method, which here is k = 12

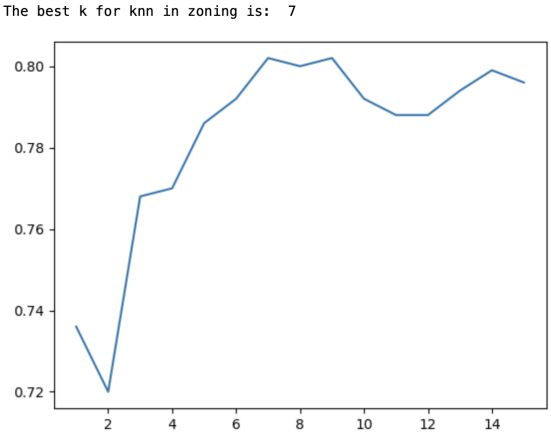

Figure 4: The diagram to find the best k in zoning feature extraction method, which here is k = 7

Now we do the same for noisy data. See Fig. 5 and 6.

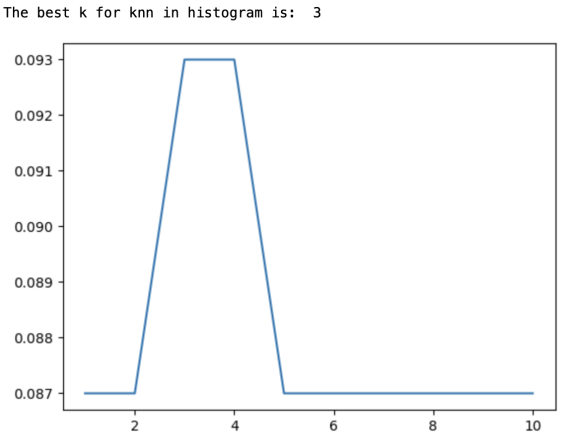

Figure 5: The diagram to find the best k in histogram feature extraction method for noisy data, which here is k = 3

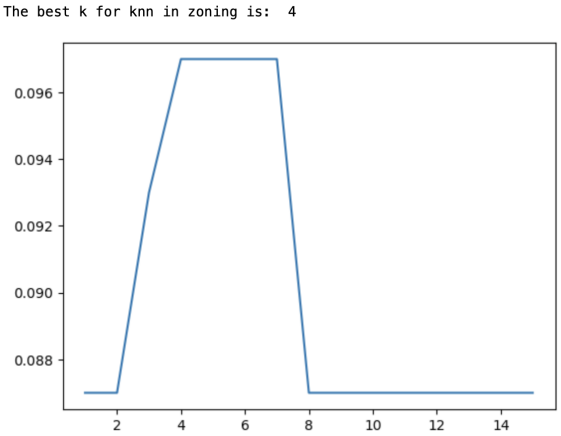

Figure 6: The diagram to find the best k in zoning feature extraction method for noisy data, which here is k = 4

8. Results

In this section, the accuracy of each pair of the four classifiers and two feature extraction techniques on the Arabic digit recognition is presented. Moreover, the KNN classification technique have a parameter that needs to be adjusted, which is discussed below.

8.1. Accuracy

Table 1 and 2 shows the accuracy of classifier/feature pairs on real and noisy data of Hoda dataset respectively.
According to the Table 1, Zoning works better than Histogram as feature extractor with the average of 75.46% . Moreover, among classifiers, Random Forest shows better result with the accuracy of 86.30 %.
According to the Table 2, both Zoning and Histogram show bad result with average of 9.5%. Moreover, among classifiers, Random Forest and Bayes show better result with the accuracy of 9.75 %. This result shows that our model does not work well for noisy data.

Feature Extractor/ Classifier	Zoning	H and V Histogram	Average
1-NN	73.60 %	37.60 %	55.6 %
k-NN	80.20 %	45.80 %	63 %
Bayes	66.00 %	42.50 %	54.25 %
Decision Tree	71.20 %	41.00 %	56.1 %
Random Forest	86.30 %	51.50 %	68.9 %
Average	75.46 %	43.68 %	

Table 1: The accuracy of classifier/feature pairs on Hoda dataset

Feature Extractor/ Classifier	Zoning	H and V Histogram	Average
1-NN	8.70 %	8.70 %	8.70 %
k-NN	9.70 %	9.30 %	9.50 %
Bayes	8.70 %	10.80 %	9.75 %
Decision Tree	9.70 %	9.70 %	9.70 %
Random Forest	10.80 %	8.70 %	9.75 %
Average	9.52 %	9.44 %	

Table 2: The accuracy of classifier/feature pairs on noisy data of Hoda dataset

8.2. *Sample Size*

In this part, we classify the data with different sample sizes from 300 to 3000 in order to find the effect of training sample size on accuracy. See fig. 7 and 8.

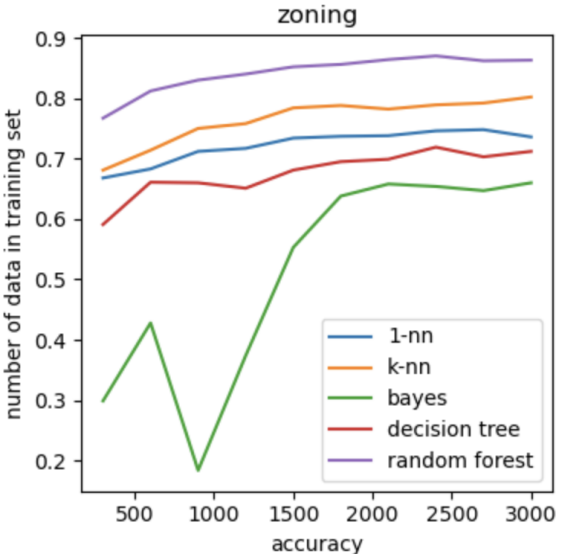

Figure 7: The effect of sample size on data with zoning feature extraction

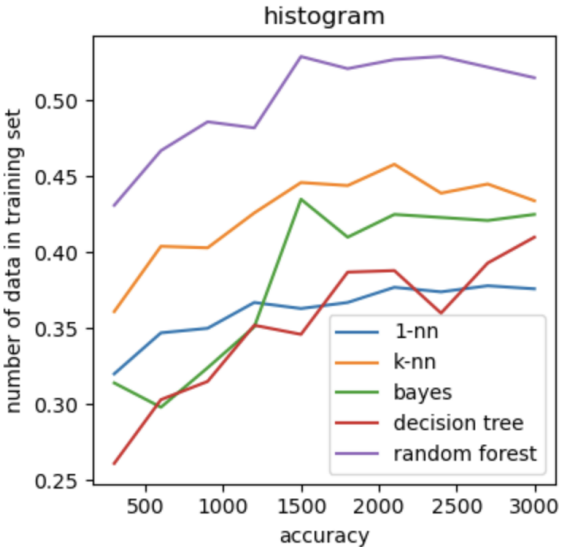

Figure 8: The effect of sample size on data with histogram feature extraction

Due to the diagrams, Random Forest performs better in overall and in zoning, bayes and in histogram 1-NN are the weakest classifiers.