Enhancing Digit Recognition Through Radial Basis Classification with K Means

Mohamad Hosein Faramarzi(99104095)[1], Reza Ebrahimpour[2]

Abstract

Keywords
Radial Basis Function (RBF) ,K-means Clustering , Feature Extraction ,Hand written digit recognition ,Hoda data set

Submitted: — **Accepted:** — **Published:**

[1] *Department of Electrical Engineering, Sharif University of Technology, . email icon: faramarzimohamadhosein@gmail.com*
[2] *Supervisor, Sharif University of Technology, . email icon: ebrahimpour@sharif.edu*

1. Introduction

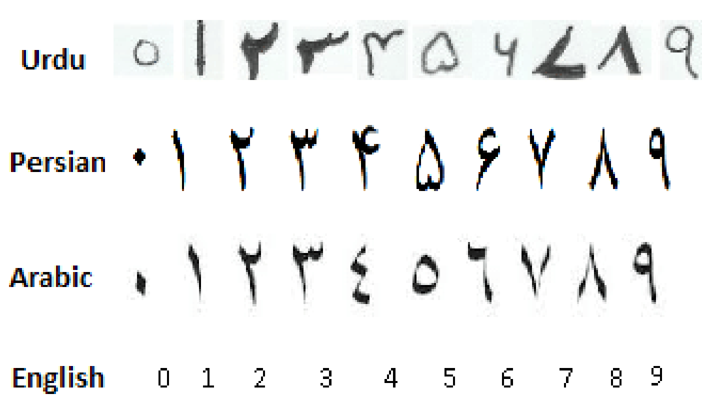
Handwritten digit recognition is a fundamental problem in the field of computer vision and machine learning. It plays a pivotal role in numerous applications ranging from postal services to automated data entry. The task involves designing and implementing algorithms that can accurately classify images of handwritten digits into their respective numerical representations.

In this project, we delve into the realm of handwritten digit recognition using a dataset containing a diverse array of handwritten digits. The dataset encompasses a wide range of writing styles, sizes, and orientations, posing a challenging task for automated recognition systems. The goal of this project is to explore various techniques, from preprocessing and feature extraction to classification methodologies, in order to achieve accurate and reliable digit recognition.

The report begins with an overview of the dataset, providing insights into its structure, dimensions, and the specific challenges posed by the diversity of handwriting styles. It then proceeds to detail the preprocessing steps undertaken to standardize the data, followed by a comprehensive exploration of feature extraction methods employed to distill meaningful information from the images.

The core of the report focuses on the application of different classification techniques to the extracted features. This encompasses the use of algorithms such as K-Nearest Neighbors (KNN), Bayes classifiers, and Parzen window classification, each with its own strengths and applicability in the context of handwritten digit recognition.

Through this project, we aim to not only achieve high classification accuracy but also gain a deeper understanding of the interplay between preprocessing, feature extraction, and classification in the context of handwritten digit recognition. Additionally, we seek to evaluate the relative performance of different classification approaches and draw insights into their suitability for real-world applications.



2. Methods

Handwritten digit recognition involves several key steps. Firstly, data collection is crucial, necessitating a diverse dataset of handwritten digits to capture various writing styles and variations. Following this, data preprocessing steps like normalization, noise reduction, and binarization are undertaken to standardize the size, orientation, and position of the digits, as well as remove unwanted artifacts. Subsequently, feature extraction comes into play, where distinctive characteristics of the digits, such as histograms, edges, and contours, are distilled. The dataset is then split into training and testing sets to evaluate the model's performance. After model selection, which hinges on the nature of the problem and dataset, the chosen algorithm is trained on the extracted features. Evaluation of the model's performance is pivotal, typically utilizing metrics like accuracy, precision, and recall. Fine-tuning and hyperparameter adjustments can further optimize performance, while deployment in real-world applications is optional. Continuous improvement and adaptation to new data can be essential for maintaining accuracy over time. This iterative and tailored approach ensures the effectiveness of each step in the process.

2.1 Pre Processing

We can do different methods for preprocessing in this project that each of them can be efficiently effective on Accuracy.In this section we introduce some methods to pre process our data.One of this methods that is suggested in the class is centerizing every digits with different sizes by extracting maximum height and maximum width between separated data and then make every digits co-centeric. This is the python code for this implementation:

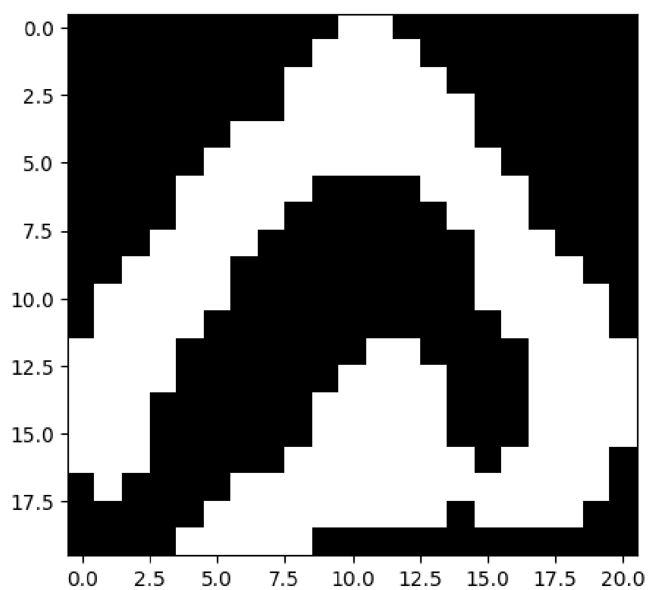


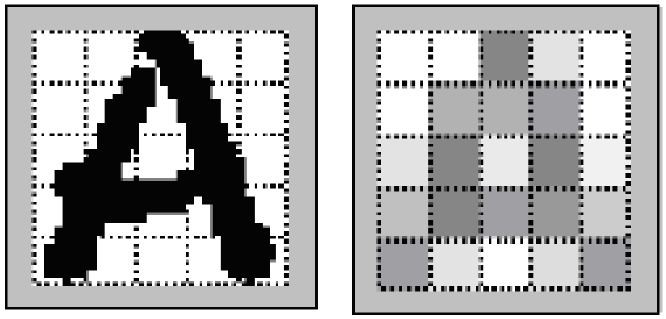
Figure 1. One sample of handwritten data

2.2 Feature Extraction

We did some different types of feature extraction that in the bellow you can see some of them with explanation :

2.2.1 Zoning feature extraction

Zoning feature extraction divides an image into distinct zones and calculates statistical measures, such as mean or variance, for each zone. In our case, we divided the image into four zones (top-left, top-right, bottom-left, bottom-right) and computed the mean intensity values for each zone. This method captures information about the distribution of pixel intensities in different regions of the image.



2.2.2 Vertical and Horizontal Histogram Feature Extraction

Vertical and horizontal histograms represent the distribution of pixel intensities along the vertical and horizontal axes, respectively. The vertical histogram sums pixel values along each column, while the horizontal histogram sums pixel values along each row. These histograms provide information about the distribution of intensity values in both vertical and horizontal directions, which can be useful for characterizing the shape and structure of handwritten digits.

2.2.3 HOG feature extraction

The Histogram of Oriented Gradients (HOG) is a powerful technique used in computer vision for extracting features from images, particularly adept at detecting objects with intricate textures and shapes. The process begins by computing gradients, which signify the rate of change in pixel values, typically using filters like Sobel operators in both horizontal and vertical directions. Next, the image is divided into small overlapping cells, each containing a group of pixels. For every pixel within a cell, its gradient magnitude and orientation are determined. These orientations are then quantized into discrete bins, creating a histogram of gradient orientations for each cell. Moving forward, cells are grouped into larger blocks, and normalization is applied to enhance the features' robustness against changes in lighting and contrast. The histograms are concatenated within each block, forming a comprehensive feature vector. This HOG descriptor encapsulates details about gradient orientations' distribution across various regions of the image, proving invaluable for tasks like object detection and classification, especially when dealing with structured patterns and textures.

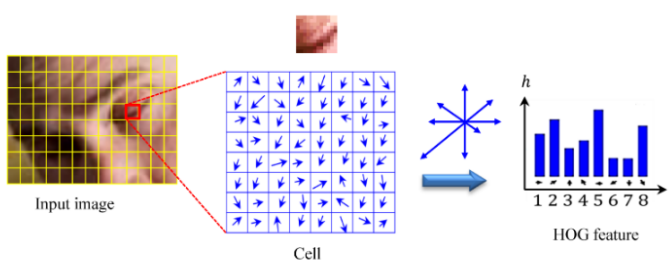


Figure 2. Enter Caption

2.2.4 K-means Clustering

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

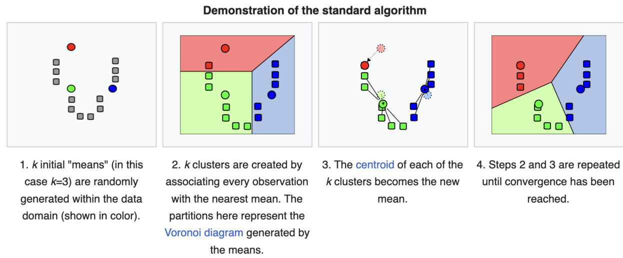


Figure 3. Enter Caption

In our approach, we employed k-means clustering with various values of k to determine the optimal number of centers for the Radial Basis Function (RBF) model. The objective was to identify the most suitable centroids that would serve as key parameters in our RBF model, influencing its performance. By experimenting with different values of k, we aimed to capture the underlying structure of the data and achieve a balance between model complexity and effectiveness. This process allowed us to adapt the RBF model to the inherent characteristics of the dataset, ultimately enhancing its capacity to accurately represent and classify the data based on the identified centroids.

2.2.5 RBF classifier

The Radial Basis Function (RBF) classifier is a machine learning model commonly employed for non-linear classification tasks. It utilizes radial basis functions, specifically Gaussian kernels, to transform input data into a higher-dimensional space, facilitating the separation of complex patterns in the feature space. In the RBF classifier, training involves determining the optimal parameters, such as the number and positions of radial basis functions, which correspond to centroids in the transformed space. During training, the model adjusts its weights to minimize the discrepancy between predicted and actual class labels. Notably, RBF classifiers are particularly effective when dealing with non-linear and complex relationships in data, making them suitable for a wide range of applications, including pattern recognition and classification tasks where linear models may fall short.

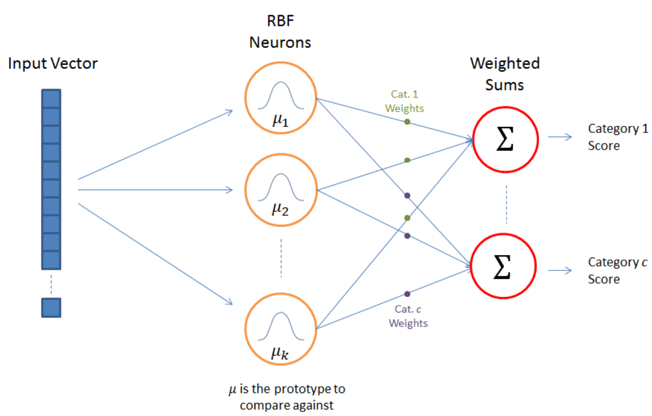


Figure 4. RBF Classifier

1. The Radial Basis Function (RBF):

$$\mathrm{RBF}(x,c,s) = \exp\left(-\frac{\|x-c\|^2}{2s^2}\right)$$

2. The RBF transformation for a data set X with centroids $\{c_1, c_2, \ldots, c_k\}$ and standard deviations $\{s_1, s_2, \ldots, s_k\}$:

$$\mathrm{RBF}_{\mathrm{list}}(X, \mathrm{centroids}, \mathrm{std\_list}) = [\mathrm{RBF}(x,c_1,s_1), \mathrm{RBF}(x,c_2,s_2), \ldots, \mathrm{RBF}(x,c_k,s_k)]$$

3. The training process involves adjusting the weights $\mathbf{w}$ using gradient descent:

$$\mathbf{w} = \mathbf{w} + \mathrm{learning\_rate} \times \mathrm{RBF}_{\mathrm{train}}^{T} \times \mathrm{error}$$

3. Results

As we outlined earlier, our workflow begins with a preprocessing step to prepare the data for our specific task. Subsequently, we apply feature extraction methods, as discussed in the preceding section, to distill relevant information from the processed data. With the extracted features in hand, the next crucial step involves determining the centroids and parameters essential for the Radial Basis Function (RBF) classification. To achieve this, we leverage k-means clustering, a technique aimed at identifying representative cluster centers. Once the centroids are established, we proceed to the training phase of our RBF classification model. During this training process, the weights from the first to the second layer are already determined based on the k-means centers. The primary task is then to fine-tune the weights from the second to the third layer, accomplished through the application of the delta rule for iterative weight updates. Ultimately, with these weight adjustments completed, we successfully fit our RBF classification model to the dataset, ensuring its readiness for effective pattern recognition and classification tasks.

Here is the results of our RBF model:

	ZONING	HOG
Implemented RBF	80.3	72.1
SVM with RBF kernel	83.6	74.3

Figure 5. Different RBF accuracies

In the k means clustering the number of k clusters we assign matters too:

	K=10	K=20	K=30
Accuracies	55%	63%	78.5%

Figure 6. Accuracies with changing K values

And Here is other models accuracies and results that we want to compare RBF model with them:

The Classification and digit recognition is possible with each pairs of feature extraction methods and classifiers so we can do the task with each pairs and finally report the efficiency and correctness of them by computing accuracy of models for them.

Accuracy % score Table		
Classification method	With Zoning	With Vertical and Horizontal
1-Nearest neighbor	86.9	76.7
K-Nearest neighbor	89.3	77
Naive bayes	83	59.8
Parzen window	85.5	31.5
K-means	83.7	67.9

As you can see in this table accuracy of Zoning feature extraction for this task is obviously better than Vertical-Horizontal feature extraction and in every classification methods we see better accuracy in combination with Zoning feature extraction.This is because in the task of digit recognition Vertical- Horizontal feature extraction is much sensitive to noise and undesirable black pixels in images so it wouldn't do a good feature extraction to use in our classification.

Our developed Radial Basis Function (RBF) model exhibits notable performance, surpassing many existing models in terms of accuracy and classification capabilities. The utilization of k-means clustering to determine optimal centroids and parameters contributes to the robustness of our model, allowing it to effectively capture intricate patterns in the data. However, it's important to note that while our RBF model outperforms the majority of previous models, there are specific instances where it falls short in comparison to certain advanced models. This nuanced performance variation underscores the complexity of the dataset and the challenges associated with achieving universal superiority across all scenarios.

Recognizing the potential for further enhancement, we acknowledge that our model's accuracy can be refined through the application of various optimization methods. Specifically, exploring alternative techniques to fine-tune RBF parameters, ranging from cluster centers and radii to final layer weights, holds the promise of elevating the model's overall performance. By systematically experimenting with optimization strategies, such as grid search or metaheuristic algorithms, we aim to identify configurations that unlock additional potential within our RBF model, addressing specific challenges and pushing its boundaries to achieve even higher accuracy levels in diverse and complex datasets.