Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 537428, 14 pages
http://dx.doi.org/10.1155/2014/537428

Research Article

A New Dataset Size Reduction Approach for PCA-Based Classification in OCR Application

Mohammad Amin Shayegan[1] and Saeed Aghabozorgi[2]

[1] *Image Processing and Pattern Recognition Research Lab, R&D Center, Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia*
[2] *Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia*

Correspondence should be addressed to Mohammad Amin Shayegan; mashaygan@siswa.um.edu.my

Received 25 August 2013; Revised 14 January 2014; Accepted 19 January 2014; Published 17 April 2014

Academic Editor: Yi-Hung Liu

A major problem of pattern recognition systems is due to the large volume of training datasets including duplicate and similar training samples. In order to overcome this problem, some dataset size reduction and also dimensionality reduction techniques have been introduced. The algorithms presently used for dataset size reduction usually remove samples near to the centers of classes or support vector samples between different classes. However, the samples near to a class center include valuable information about the class characteristics and the support vector is important for evaluating system efficiency. This paper reports on the use of Modified Frequency Diagram technique for dataset size reduction. In this new proposed technique, a training dataset is rearranged and then sieved. The sieved training dataset along with automatic feature extraction/selection operation using Principal Component Analysis is used in an OCR application. The experimental results obtained when using the proposed system on one of the biggest handwritten Farsi/Arabic numeral standard OCR datasets, Hoda, show about 97% accuracy in the recognition rate. The recognition speed increased by 2.28 times, while the accuracy decreased only by 0.7%, when a sieved version of the dataset, which is only as half as the size of the initial training dataset, was used.

1. Introduction

The emergence of the Big-Data issue has caused researchers to focus their attention on the size reduction and also dimensionality reduction of the data to save time and memory usage. Also, there is an increasing demand for employing various applications on limited-speed and limited-memory devices such as mobile phones and mobile scanners [1]. In this context, there is a pressing need to find efficient techniques for reducing the volume of data in order to decrease overall processing time and memory requirements.

A survey of the literature on large dataset issue reveals that two general approaches are used for dataset volume reduction—dimensionality reduction and size deduction. In the dimensionality reduction technique, the system will try to find and remove less important extracted features from the dataset samples. These techniques are widely employed in different areas such as EMG signal feature reduction [2] and gene expression dataset reduction [3]. Specific examples of these techniques include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Random Projection (RP), and Mean-Variance Approach [4, 5]. However, finding an optimal, effective, and robust feature set from a big initial extracted feature is usually a heuristic and inevitably a difficult task [6].

In the size reduction techniques, the system will try to reduce the number of objects or observations in a dataset. Such techniques find and remove two groups of samples from a dataset—samples far from a class centroid (outlier samples or support vector samples) [7–9] and samples near to each class centroid (e.g., using K-means clustering technique) [8, 10]. However, the samples near to a class centroid include important information about various characteristics of a class, and they are necessary to make the system model.

Also, the outlier and support vector samples are necessary to evaluate the system efficiency and functionality.

Currently, large amounts of indispensable data are available on paper. Converting the graphical-text data images into editable text documents (Optical Character Recognition (OCR)) is one of the most attractive branches in the pattern recognition (PR) domain that scientists have been faced with in recent years. Meanwhile, about more than one billion people worldwide use the Arabic, Farsi, and other similar alphabets as their native language [11]. About 30% of the world population and around 30 world languages use this group of alphabet sets as a base script for writing, and this makes it the second most-used alphabet set used in writing worldwide [12]. Hence, it is not surprising that these alphabets have received a lot of attention in the recent years. Concerning dataset size reduction in OCR applications, the PCA technique has been used to compress the features space in the numeral part of the CEDAR database [13], the MNIST database [14], and also the handwritten Tamil characters classes [15]. However, to the best of our knowledge, no research efforts have been undertaken regarding dataset size reduction for the Arabic/Farsi language.

This paper presents a new method to achieve the above purpose by using a Modified Frequency Diagram (MFD) matching technique and a new similarity measurement function, in particular, to reduce the number of samples in the training section of an Arabic/Farsi dataset. Using the MFD technique, a template is first generated for each class in the pattern space. A new similarity measurement function is then used to compute a similarity value for each training sample of a dataset corresponding to that class template. Thereafter, the samples in a specific class are sorted in descending order based on the calculated similarity values. Finally, the sorted training dataset is sieved at sampling rates of 1/2, 1/3, and 1/4, and the dataset size is reduced to 1/2, 1/3, or 1/4 of its original size. To investigate the efficiency of the proposed dataset reduction technique, the sieved training dataset along with PCA is used in the recognition stage of an OCR application to recognize handwritten Arabic/Farsi numerals. The results from a successful application on an OCR database have been reported, but what is more important to note is that this method, with some adjustment, can be used for other pictorial datasets.

The main difference between our proposed method and other available methods for dataset size reduction is that our model keeps some samples which are close to a class template, as well as those which are far from a class template, in the final reduced dataset version. Hence, it is possible to have appropriate samples for making a system model and also for assessing the system efficiency with a high degree of accuracy. Moreover, in this research, PCA is employed both for automatic feature extraction and feature reduction (selection) in OCR applications.

This paper contributes to the corpus of knowledge in dataset size reduction as follows: (1) it proposes a new method for dataset size reduction to speed up system training and testing and (2) it is the first successful effort to be reported on the use of PCA for automating feature extraction, in addition to reducing dimensionality (feature reduction) in OCR applications.

This paper is organized as follows. Section 2 discusses the background of the research topic and introduces works related to the topic. Section 3 presents the proposed size reduction technique, the research method, and the experimental procedures. Section 4 presents the results of the experiments and the analysis, and finally, Section 5 concludes the paper.

2. Background and Related Works

2.1. Principal Component Analysis (PCA). The Principal Component Analysis (PCA) technique is a classical statistical linear transform which has been widely used for different PR applications such as data compression, face recognition, and character recognition [6]. It has been applied to find important patterns in high-dimensional input data. It converts a correlated feature space into a noncorrelated feature space. In the new space, features are reordered in decreasing variance values—based on the generated eigenvectors from the training data—such that the first transformed feature accounts for the most variability in the data.

PCA can briefly be described as follows. Let B be a $N \times N$ pixels binary image as a random vector population. A 2D image B is converted to a 1D vector X by concatenating all rows of the image in order as follows:

$$X = \left(x_1, \ldots, x_n\right)^T, \tag{1}$$

where X_i is the pixel values of row i. If μ_x is the mean of X, then the covariance matrix of that population is C_x:

$$C_x = E\left\{\left(x - \mu_x\right)\left(x - \mu_x\right)^T\right\}. \tag{2}$$

In order to normalize the data values in each dimension, data is subtracted from the corresponding mean. This changes the mean of each dimension to zero. The components of C_x represent the covariance between the random variable components x_i and x_j. The covariance matrix C_x is a square matrix, and therefore its eigenvalues and eigenvectors can be calculated. The eigenvectors are generally perpendicular to each other. The eigenvector with the biggest corresponding eigenvalue is the most significant representative data and is considered to be the first most significant principal component. The eigenvector with the second biggest corresponding eigenvalue is considered to be the second most significant principal component, and so on. Therefore, by sorting the eigenvalues in descending order, the most important eigenvectors as the most significant representation data are found. In this way, the pattern space dimension can be reduced.

However, the computation of PCA requires eigenvalue decomposition of the covariance matrix of the feature vectors with around $O(d^3 + d^2 n)$ computations, where n is the number of samples and d is the dimensionality of the feature space [16]. This powerful technique, however, is time consuming and is usually employed for feature extraction/selection operations on small scale datasets.

2.2. Feature Extraction and Feature Selection. Feature extraction (FE) is a task to detect and extract the maximum amount of the desired attributes from the input data. Features are the information that is fed to the recognizer to build a system model [17]. They should be insensitive to irrelevant variability in the input as much as possible, should be limited in number to permit effective computation of discriminant functions, and should not be similar, redundant, or repetitive. Usually, extracting appropriate and robust features is a difficult task in an OCR system, like in other PR applications.

Various features are computed in the feature extraction module in an OCR system. The features are categorized into global transformations such as Fourier, structural features such as ascenders and descenders, statistical features such as moments, and template matching and correlation.

Many different features can be found or calculated for each pattern in a PR system. Some of the features, however, might correspond to very small details of the pattern, or some of them might be a combination of other features (nonorthogonal features), while others might not play any effective role in the recognition stage. Irrelevant or redundant features may degrade the recognition results and significantly reduce the speed of the learning algorithms. Hence, using all extracted features does not always produce the desired results and could also increase the time complexity of the recognition process [18]. Therefore, following the feature extraction process, another important process—feature selection (FS)—is involved and this process can reduce the problem dimensionality. FS is typically a searching technique for finding an optimal subset with m features out of the original M features.

The first category of feature selection methods is Sequential Backward Selection (SBS). In this approach, features are deleted one by one and system performance is measured to determine the feature performance. However, it is very important to find the correct order of deleting the features one by one. This is because the derived efficiencies of a system after deleting features A, B, and C are not the same as its derived efficiencies after deleting the features in the order A, C, and B, or B, C, and A, and so on [19].

The second category of feature selection methods comprises the random search methods such as Genetic Algorithms (GA). The GA methods select chromosomes (features) with the best recognition percentage, one by one, and move this chromosome to the next stage. However, it is possible that when a good characteristic feature is combined with another feature, the overall performance will not be as good as the performance of each of the features, separately. Azmi et al. [18] used GA in an OCR system for recognizing handwritten texts. Initially, there were 81 features and the recognition rate was 77%. After applying GA, the number of features was reduced to 55 and the recognition rate improved from 77% to 80%. Kheyrkhah and Rahmanian [20] employed GA to optimize the number of initial extracted features in a recognition system for handwritten digits. They found that all the extracted features are not useful in classification, and they also reduce recognition accuracy and increase the system's learning time. Their system was able to reduce the number of features from 48 to 30 and increase the recognition rate from 75% to 94%, but the elapsed time for these improvements was significant.

The third category of feature selection methods is represented by the Principal Component Analysis (PCA). PCA transforms data into a new space where the features are uncorrelated. In the new space, features are reordered in decreasing variance value such that the first transformed feature accounts for the most variability in data. Hence, PCA is able to overcome the problem of high dimensionality and colinearity [6]. It is obvious that handwritten digits and characters have a wide variety of writing styles; hence, handwritten texts are placed in the high-dimension input space category. It means that PCA can be an effective tool for attribute reduction in OCR applications.

2.3. Dataset Size Reduction. There have been researches on finding methods to reduce dataset size in order to decrease the overall processing time in PR systems. Urmanov et al. [21] first calculated an original decision boundary equation D for each class of patterns. They then calculated the distance between each sample x in each class C and the original decision boundary equation D. Next, without using sample x in the same class C, they calculated a new decision boundary and the new distance with respect to this new decision boundary. If the pairs of old and new decision boundaries are very similar, sample x is considered worthless and is removed from the dataset.

Zhongdong et al. [7] attempted to reduce dataset volume by finding support vectors. They found the samples of each class near the boundary spaces and then calculated the distance of the found samples of each class from other classes. Finally, they considered the nearest couple of samples from any two classes as the most important, while the other samples of each class were removed from the classes and dataset. This approach was also used by Hara and Nakayama [9].

Vishwanathan and Murty [8] first used multicategory proximal SVMs to categorize training system prototypes and then found different boundaries for separating different classes of clusters. They removed not only samples which are generally close to the class boundaries but also the typical patterns that are far from the class boundaries.

For each pair of samples from two different classes, Javed et al. [22] plotted a sphere for each pair of samples, such that those two samples are put on two sides of the sphere diameter. If none of the other samples inside these two classes are within the sphere volume, these samples are nearest to each other from these two classes. Therefore, these samples are support vectors and will be inserted into the final dataset.

Boucheham [23] introduced the method of recursive Piecewise Linear Approximation (PLA) and sequential PLA for data reduction to speed up time series comparison. It is usually possible for printed characters to be considered as a time series, but the wide variation of styles of handwritten characters rules out any possibility of considering handwritten template as time series, and therefore the method mentioned is not applicable for handwritten documents.

Cervantes et al. [24] first obtained a sketch from the distribution of available classes with a small number of training samples, and they then identified existing support vectors in this limited dataset. Their proposed system is trained to find samples near the boundary between classes, and then other important samples are found and added to the final dataset.

In summary, it is found that the majority of the algorithms mentioned can be divided into two general groups as follows.

(a) The first group of algorithms tries to find and delete support vector samples (SVs) in all classes [8]. These samples are usually the samples which are far from the class center and near to a class boundary. It is usual for a recognition system to justify or classify such samples wrongly. However, one of the main criteria for evaluating system efficiency and measuring the power of a PR system is the correct recognition of these SVs and outlier samples. Hence, it is not a good strategy to delete these patterns from the initial dataset in order to achieve dataset size reduction.

(b) The second group of algorithms removes the samples near the centers of the classes from the initial training dataset to create a short final dataset version [7–9, 22, 24]. However, these samples include highly valuable information about a specific class that is needed for making a system model in the training phase of a PR system.

2.4. PCA-Based Classification. The Principal Component Analysis (PCA) technique is usually used for feature reduction, but sometimes it is also utilized for automatic feature extraction operation in various PR systems.

Zhang et al. [14] introduced a multimodal approach for reducing the features' dimensions in an OCR system and tested their approach on the numeral part of the MNIST dataset. They employed PCA for feature compression and succeeded in reducing the CPU time for classification. Kim et al. [13] modeled each digit class of the UCI dataset to several components and used a mixture model of PCA techniques to move extracted components into a decorrelated feature space. Based on the results obtained, the PCA mixture model outperforms other methods such as k-NN in terms of accuracy. In 2004, Deepu et al. [15] employed PCA for dimensionality reduction in an online OCR application for Tamil handwritten characters. They found that the modified version of the orthogonal distance classifier performs better when compared to the k-NN classifier. The novelty of this work is that it is language independent, and the proposed method can also be used for other language scripts. In 2005, Mozaffari et al. [25] used PCA technique to reduce the number of fractal code features to 240 in a handwritten zip code recognition system. They succeeded in improving the recognition rate from 86.3% to 90.6%. Ziaratban et al. [26] extracted a set of feature points—including terminal, two-way, and three-way branch points—from the skeletons of characters. Finally, each skeleton was decomposed into some primitives, which are curved lines between any two successive feature points. Since the number of primitives varies from character to character, they used a PCA algorithm to reduce and equalize the length of the feature vectors. They achieved 93.15% recognition rate using a dataset with 7,647 test samples. To recognize handwritten Arabic isolated letters, Abandah et al. [6] extracted 95 features from input images. The PCA technique was then used for feature reduction, and only the first 40 features were selected. They achieved an average accuracy of 87%, in the best case. El-Glaly and Quek [19] extracted four sets of features, S1, S2, S3, and S4, from input data, for the use in an Arabic OCR system. They trained the system with the four feature sets, separately. These sets were then processed using the PCA algorithm, and PCA rearranged the features based on their importance which was identified by the recognition system. The results show that feature X in rank 23 in set S3 took rank 7 in set S1, and so on. The results of experiment indicate that if feature X is deleted merely to achieve feature reduction, it may give rise to serious errors in the final results.

2.5. Farsi OCR. This research was conducted specifically on the use of OCR technique for recognition of handwritten Arabic/Farsi digits. From a review of the relevant literature, some of the methods for recognizing handwritten Arabic/Farsi numerals are described briefly in the following sections.

Mowlaei and Faez [27] extracted a 64-dimensional feature vector of Harr wavelet coefficients for recognizing handwritten Farsi digits. They achieved 92.44% recognition rate of 3,840 digits by using an SVM classifier with an exponential RBF kernel in one-to-other mode.

In 2003, Sadri et al. [28] used SVM classifiers for recognizing the digits part of the Farsi-CENPARMI database. They obtained four different views for any image from four main directions by counting the number of background pixels between the border and outer boundary of any image, and finally they created a 64-dimensional feature vector for each image. Using SVMs with RBF kernel, a 94.14% recognition rate was achieved. To compare the classifiers, they employed an MLP-NN classifier with two hidden layers. This classifier achieved an accuracy rate of 91.25%, and this is lower than that obtained using the SVM classifier.

In 2004, Mozaffari et al. [29] proposed a new method for recognizing handwritten isolated Farsi characters and numerals for handling mail codes and city names in the post ministry of Iran. They extracted a 64-dimensional set of fractal codes, as well as wavelet transform coefficients. They employed an SVM with an exponential RBF kernel as a classifier. The final results for both feature sets were almost similar and a recognition rate of 91.3% was achieved. However, the system recognition speed, when it uses wavelet coefficients as features, is 25 times higher than the system which uses fractal codes as features.

Soltanzadeh and Rahmati [30] used the outer profile of digit images at multiple orientations—top, down, left, right, diagonal, off-diagonal, and so on—as the main features. They also used the normalizing crossing counts and projection histograms of any image as complementary features. The total number of features was $32 * n + 1$, where n is the number of orientations for calculating the outer profiles. They

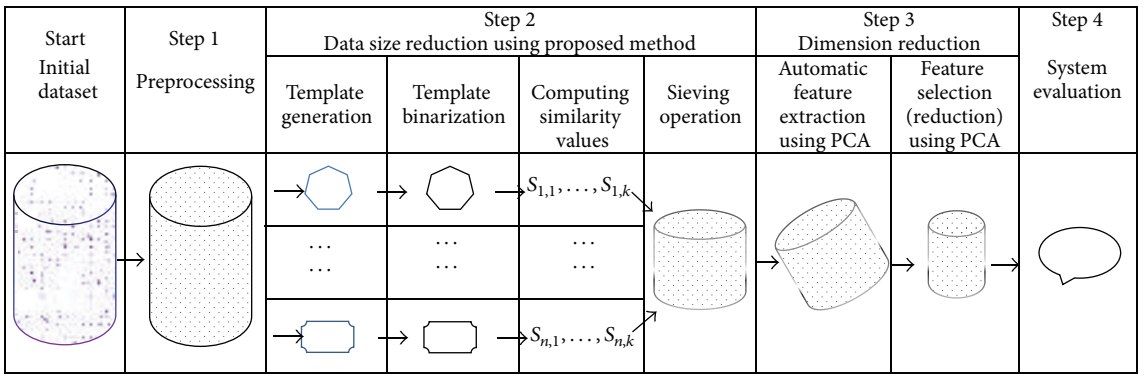


Figure 1: The proposed model for dataset (size and dimension) reduction.

employed an SVM classifier with a polynomial kernel in one experiment, and an RBF kernel in another experiment in the one-rest method. The best result they achieved was 99.57% recognition rate using eight orientation profiles and the RBF kernel.

Ziaratban et al. [31] extracted some language-based features including the position and the amount of the best-occurred matching in the horizontal and vertical coordinates for handwritten Farsi digit recognition. They tested the system on a database which contains 6,000 samples for training and 4,000 samples for testing, using an NN-MLP as a classifier, and successfully achieved 97.65% recognition rate.

Enayatifar and Alirezanejad [32] categorized images of digits into two groups: the first group includes digits 1, 2, 3, 4, 6, and 9 and the second group includes digits 0, 5, 7, and 8, based on the similarity of their skeletons. In the feature extraction stage, they divided the image of a digit into 24 frames and for each frame they calculated pixel accumulation and direction as features. Hence, a 48-dimensional feature vector was created for each digit. For the recognition operation, an MLP-NN classifier with 50 neurons in the hidden layer was employed. The best recognition rate achieved was 94.30%.

3. The Proposed Method

The existing methods already proposed for dataset size reduction generally try to find the boundary points (support vectors) between different classes in a pattern space. However, our proposed method adopts a completely different approach. Figure 1 depicts the general structure of the proposed model. The operations mentioned are described in the following sections.

3.1. Preprocessing. The performance of an OCR system depends very much upon the quality of the original data. In this context, we took into consideration that the proposed algorithm should be nonsensitive with respect to the scaling, rotation, and transformation of patterns. Hence, some important preprocessing operations, such as noise removal, dimension normalization, and slant correction using common powerful techniques, are first performed on the samples.

We applied a median filter with a 3×3 window and also morphological opening and closing operators using dilation and erosion techniques for noise removal. The image size was normalized without making any changes to the image aspect ratio, and as a result, the width or height (or both) was changed to 50 pixels and the image was located in the center of a 50×50 pixels bounding box.

The body in every Arabic/Farsi digit is constructed using only one component. Thus, after the preprocessing operations, if it is found that there is still more than one group of connected pixels in the image of the digit, the extra blocks are considered as noise or separate components of the initial image. To find and remove the rest of the noise, the pen width is estimated using three different methods, and then the average of those values is considered to be the final pen width. To achieve this, we compute the following:

(a) the mode of the image vertical projection,

(b) (the value of image density)/(the number of image skeleton pixels),

(c) {(the value of image density)/(the number of image outer profile pixels)} $*$ 2.

The results from the experiments show that the average of three values is a more accurate estimate of the pen width than each of the values alone. After finding the pen width, all small components with a pixel density that is less than two times of the pen width can be considered to be noise and can be deleted from the input image. The threshold 2 was obtained experimentally. The rest of the connected components are considered as broken parts of the digit image.

In order to connect the broken image segments together, we used a new approach. By using connected component analysis, we named the biggest available part as the main part **M** of the image. The outer contour of the main part **M** was then extracted and the coordinates of its pixels were saved in array **MAIN**. Thereafter, for all of the rest secondary components $\mathbf{S}_i$ (which are smaller than the main part **M**), we found the outer contour and saved the pixels coordinate of those outer contours in another array **SEC**. Then, we computed the Euclidean distance between all elements of array **MAIN** with all elements of array **SEC**. The smallest value of the computed distance indicates the shortest path

```
while (there is another secondary component in input image) do
{
    find outer contour of the main part M;
    save the pixels coordinate of M in array MAIN;
    repeat
    {
       find outer contour of an image secondary part S;
       save the pixels coordinate of S in array SEC;
    } until (there is not another secondary parts in image);
    for (each pixel A in array MAIN)
    {
        for (each pixel B in array SEC)
        {
           compute the distance d between pixels A and B;
           save (d, coordinate of pixel A, coordinate of pixel B) in array D;
        }
    }
    d_min = smallest value d in array D;
    A_min = coordinate of pixel A, corresponding to d_min;
    B_min = coordinate of pixel B, corresponding to d_min;
    draw (a straight line with pen_width thickness from A_min to B_min);
}
```

Algorithm 1: The proposed procedure to connect the broken image segments.

between contour **M** and one of the secondary contours $\mathbf{S}_k$. Finally, we drew a line with thickness equal to estimated pen width along the shortest path between **M** and $\mathbf{S}_k$. As a result, the main part **M** is connected to a secondary part $\mathbf{S}_k$. This process was repeated until there is not another secondary component. A new version of main part **M** is used in each iteration of the algorithm, because one secondary part is connected to the old version of main part **M** to make the new version of main part. Algorithm 1 demonstrates the pseudocode for this process.

We applied the proposed method on the digits part of the Hoda dataset [33] to connect the broken parts of the images of the digits. The results were encouraging as we were able to achieve 97.16% successful connections. Figure 2 shows an example of the above-mentioned preprocessing operations on two sets of digits from our training dataset.

The method proposed by Hanmandlu et al. [34] was used to correct the slant angle of each image. First, an image is divided into upper and lower halves. The centers of mass points for these two parts are then calculated. The slope of a line which connects these two mass point centers is considered to be the slant angle and the image is rotated in the reverse direction of this value. Figure 3 shows an example of slant correction.

3.2. Data Size Reduction

3.2.1. Template Generation for Each Class. By using all the preprocessed samples of each class, the Frequency Diagram (FD) (density grid) was first computed by calculating the number of occurrences of pixel "1" in the coordinates (x, y) for all available samples in a special class [35]. The modified version of FD (MFD) was defined by Khosravi and Kabir [33]. They increased the pixel density variable D_i by 1 unit, if the pixel in coordinate (x, y) of an input sample is "1." In order to have a more accurate description of pixel density in each class, they decreased the pixel density variable D_i by 1 unit, if the pixel in coordinate (x, y) of an input sample is "0." Equations (3) and (4) are the formulas for calculating the FD and MFD, respectively:

$$D_i(x, y) = \sum_{n=1}^{N_i} \left(F_n(x, y)\right), \tag{3}$$

$$D_i(x, y) = \sum_{n=1}^{N_i} \left(F_n(x, y) * 2 - 1\right), \tag{4}$$

where x, y are coordinates of different pixels of any sample, $x = 1, \ldots, k$, $y = 1, \ldots, p$, k, p are dimensions of normalized samples, n is the nth sample of a specified class, $F_n(x, y)$ is the pixel value of the nth sample at coordinate (x, y), and N_i is the total number of samples in class C_i.

Figures 4 and 5 show examples of FD and MFD for digit "ץ" (digit "7") in the Arabic/Farsi digits set, obtained by using (3) and (4), respectively. For simplicity, only 200 samples of digit "7" were used for generating these figures. The reason for employing the MFD approach (4) to find similarity values in our research is that the MFD concept provides a more accurate description for similarity/distance between an image and a class template. Also, the MFD matrixes are considered as templates for different classes, called Template Matrices (TMs).

3.2.2. Template Binarization. The generated Template Matrices (TMs) include a considerable amount of information

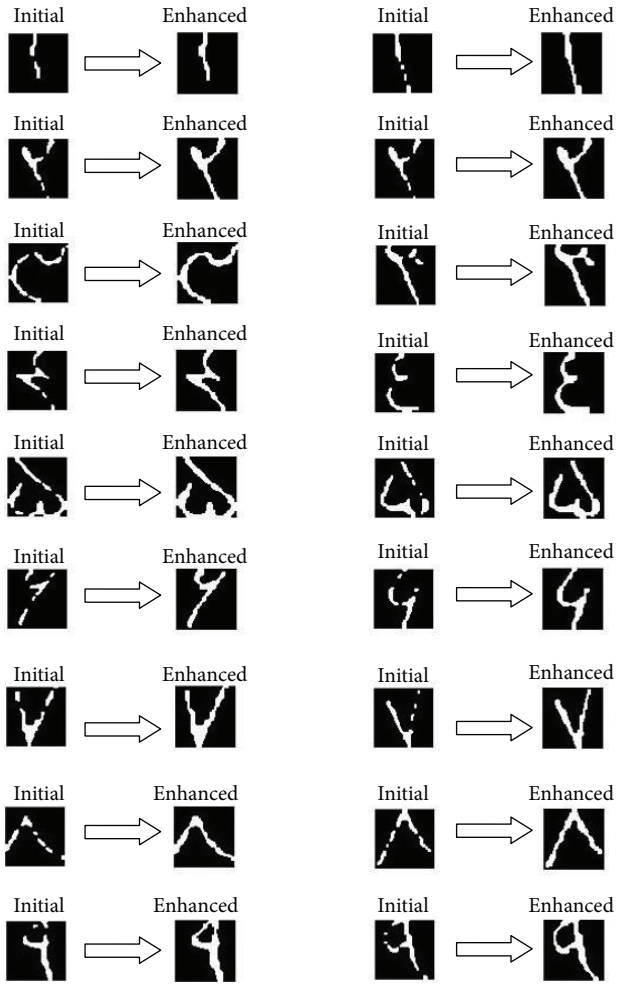


Figure 2: Applying preprocessing operations on Arabic/Farsi digits.

about the distribution of the pixels in each class. Hence, they can be considered to be a source for feature extraction in a PR system that uses the template matching technique.

The elements of a Template Matrix are computed by using MFD equation (4). Hence, they have a value between $-N_i$ and N_i. In order to compare the samples of a class with the derived template for the same class, the calculated numerical values obtained using (4) must be converted to binary numbers 0 and 1. In other words, it is necessary to convert the Template Matrices (TMs) to the Binarized Template Matrices (BTMs) version. This operation involves two steps. In the first step, we scaled the values of the TM elements to the gray levels spectrum from 0 to 255 by using (5), where N_i is the total number of samples in class C_i. In this equation, P is the initial value of each pixel in a TM:

$$\text{Gray\_Level\_TM} = \frac{(P + N_i)}{2} * \frac{255}{N_i}. \tag{5}$$

In the second step, the Gray_Level_TM elements are converted to the BTMs version by using the standard global Otsu's method. Figure 6 shows the BTM related to the template obtained in Figure 5 using the above-mentioned method.

3.2.3. Computing Similarity Value. The use of template matching techniques involves determining similarities between an input instance and all generated class templates. There are various approaches for measuring the similarity between an input data and a class template, some of which include nonmetric cosine [35], conventional definition, modified frequency density [33], Hamming distance, linear correlation, cross correlation [36], Sawaki measure, and Rogers-Tanimoto measure. Various other definitions for similarity measures and distances that have been described in the literature include Minkowski distance (L_p distance), Bottleneck distance, Hausdorff distance, fringe distance, Turning Function distance, Frechet distance, Reflection distance, and Transport distance (Earth Mover's distance) [37].

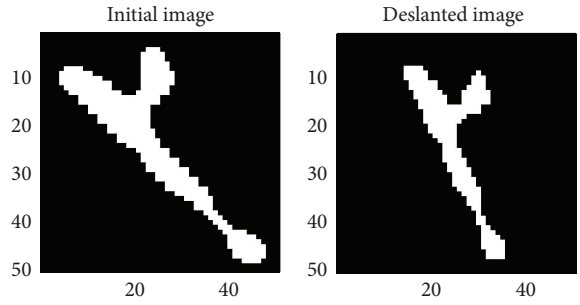
Using the Modified Frequency Diagram (4), we defined a similarity variable **S**. Similarity variable $\mathbf{S}_{k,i}$ indicates the similarity value between the kth sample of class i and the corresponding class template. It is increased by the value of the MFD_i, if an image pixel and its corresponding template pixel have the same value of "1" or "0"; otherwise, $\mathbf{S}_{k,i}$ is decreased. Also, in order to amplify the effect of similar pixels in comparison to nonsimilar pixels, we considered the effect of the equal pixels to be twice that of nonequal pixels, and we defined the general form of this new concept as follows (while the **w** reward coefficient is set to 2):

$$\begin{aligned}\mathbf{S}_{k,i} = \sum_{x=1}^{n}\sum_{y=1}^{m} \{\mathbf{w} * [f_{k,i}(x,y) \odot \text{BTM}_i(x,y)] \\ - [f_{k,i}(x,y) \oplus \text{BTM}_i(x,y)]\} \\ * \left|\text{MFD}_i(x,y)\right|,\end{aligned} \tag{6}$$

where i is the class number in pattern space, n, m are image dimensions, **w** is the reward coefficient (in this research, this parameter was set to 2), $f_{k,i}$ is kth sample image of class i, BTM_i is ith Binarized Template Matrix (corresponding to class C_i), MFD_i is ith Modified Frequency Diagram matrix (corresponding to class C_i), $\odot$ is logical XNOR operator, and $\oplus$ is logical XOR operator.

Experimental results show that the calculated values for similarity variables $\mathbf{S}_{k,i}$ proposed in (6) have wide variances, and this leads to better differentiation between the samples in a class. We set the **w** reward coefficient in (6) to (2). When **w** coefficient is increased, it will increase the effect of similar corresponding pixels in calculating the similarity values. It must be noted, however, that choosing too big a value for **w** will cancel the fine effect related to the corresponding nonsimilar pixels in (6).

3.2.4. Sieving Operation. The proposed similarity equation (6) was applied to the training dataset and a similarity value was computed for every sample in each class in order to increase overall system speed. Finally, all the training samples in a particular class were sorted in descending order based on their computed similarity values.



(a) Initial shape (b) Shape after slant correction

FIGURE 3: Shape of digit "۲" ("2") before and after slant correction.

FIGURE 4: Frequency Diagram (FD) matrix for Arabic/Farsi digit "۷" ("7") using (3).

The original dataset in this research contains 60,000 samples. By performing sampling at the rates of 1/2, 1/3, and 1/4, using the sorted version of the training dataset based on computed similarity values, three reduced training dataset versions were finally kept—half (30,000 samples), one-third (20,000 samples), and one-fourth (15,000 samples). These three reduced versions of the training dataset, as well as the initial training dataset, were used in this research.

3.3. Dimensionality Reduction. In the aforesaid literature, it is found that PCA has been mostly used only for feature selection (dimensionality reduction). In this study, however, we have applied PCA not only for feature reduction but also for automatic feature extraction.

3.3.1. Automatic Feature Extraction Employing PCA. Because of the wide variations in writing styles, handwritten characters are put in the high-dimension data category and the order of the pixel values of a binary image can be considered as a random vector population. Hence, in the first step of the automatic feature extraction operation, a preprocessed training image was rescaled to 20 × 20 pixels image, and it was then converted to a 1D vector including 400 (20 × 20) elements. Following this, the generated pixel-based vectors of all training samples were fed directly to the PCA algorithm in order to calculate their eigenvalues and eigenvectors. The output of this stage was a new description of an image with 400 coefficients related to eigenvectors. These new coefficient vectors were considered as the initial features set of the input images.

3.3.2. Feature Selection (Reduction) Utilizing PCA. After automatic feature extraction, feature selection was carried out. The reduced version of the feature vectors was produced using the h first significant eigenvector ($h \leq 400$) from the initial 400-dimensional feature vector created by PCA.

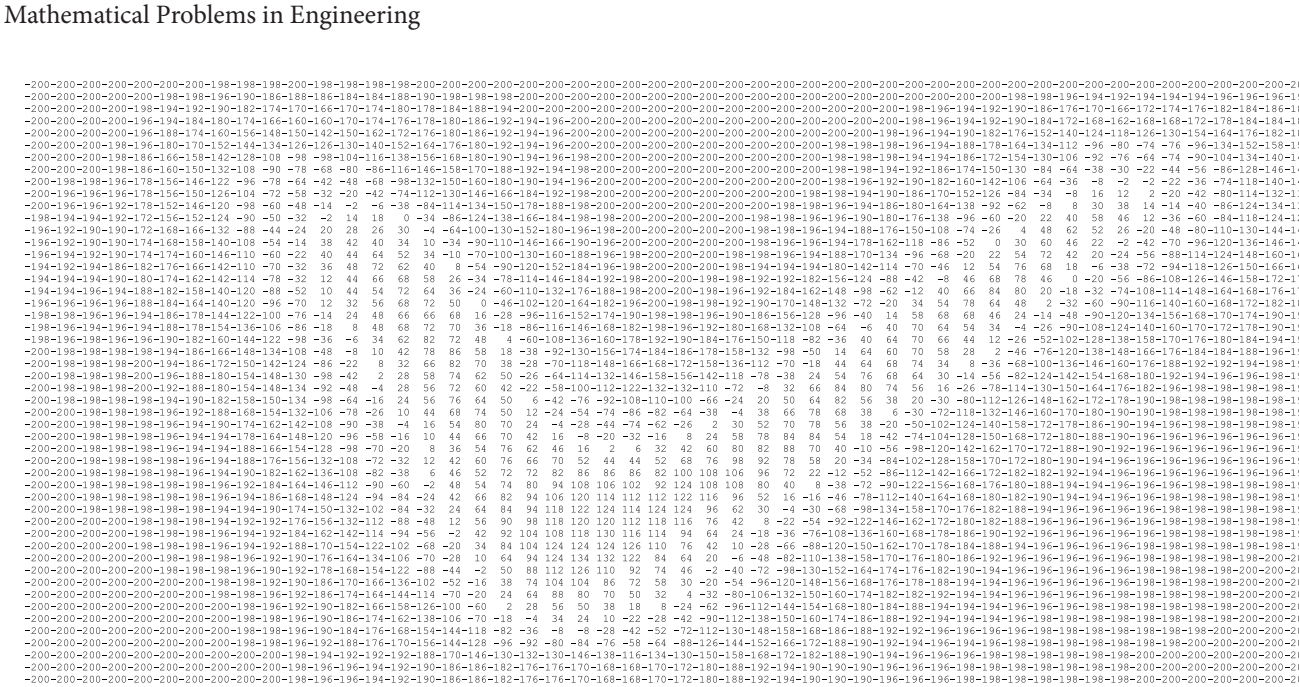


Figure 5: Modified Frequency Diagram (MFD) matrix for Arabic/Farsi digit "۷" ("7") using (4).

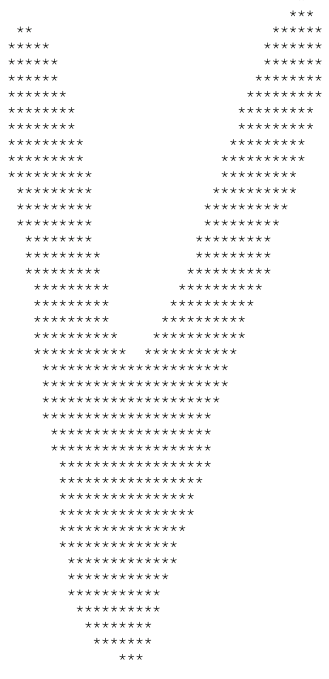


Figure 6: Binarized Template Matrix (BTM) corresponding to Figure 5.

4. Experimental Results

4.1. Hoda Dataset. All PR systems need initial datasets for training the system. In recent years, researchers have produced some benchmark datasets in order to encourage more researches in the PR domain and also to compare the functionalities of various PR systems that have been developed.

This research was conducted using the Arabic/Farsi OCR datasets and more specifically on handwritten Farsi/Arabic texts. Some available handwritten Farsi standard datasets with digits section include IFHCDB, Hoda, CENPARMI, and Hadaf [38]. Similar datasets for handwritten Arabic alphabets with digits part are Al-Isra, CENPARMI, ARABASE, and LMCA [39].

The digit part of the Hoda dataset [33]—one of the largest Farsi (and also Arabic) handwritten standard datasets—was chosen to test the proposed method. The Hoda dataset has two sections digits and characters. The digit section was prepared in 2007 by extracting the images of digits from 11,942 application forms for university entrance. Those forms were scanned at 200 dpi in 24-bit color format. The digits were extracted from the *postal code, national code, record number, identity certificate number,* and *phone number* fields of each form. The digit section of the Hoda dataset contains 80,000 samples and is divided into two parts—60,000 training samples and 20,000 testing samples. Figure 7 shows some sample digits from this dataset.

4.2. Proposed Method. In this research, the same operations were carried out in the preprocessing step on the training and testing samples. The outputs were noise-filtered, reslanted, relocated, and dimension-normalized. To save time and memory requirement, we rescaled all the preprocessed images into 20×20 pixel images again.

One template was created for each class by using all the training samples in the training part of the Hoda dataset and the Modified Frequency Diagram in (4). The templates were binarized using Otsu's method. The PCA technique was applied on the training dataset to extract more important features automatically. For this task, all 400 pixels of any images were fed directly to the PCA algorithm. The data

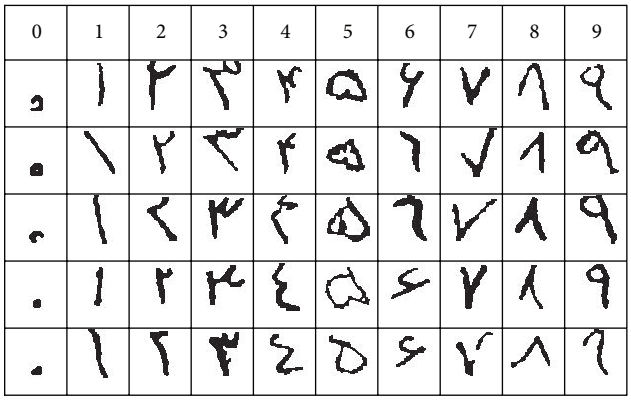


Figure 7: Some sample digits in the Hoda dataset.

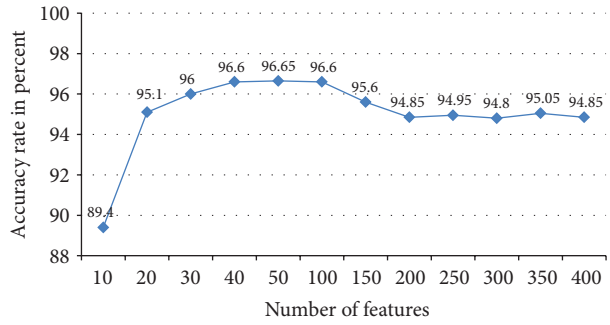


Figure 8: Accuracy versus the number of features proposed by PCA.

was then mapped into the new orthogonal space based on the derived eigenvectors. Following that, we employed the proposed similarity function (6) and computed a similarity value for each training data sample. Based on these computed values, the training data in each class were sorted in descending order. Using sampling operation, the sorted dataset was sieved into 1/2, 1/3, and 1/4 of the original dataset volume for being used in further experiments.

It is very important to select a good classifier for a PR system. For this reason, some researchers use a combination of classifiers to achieve better results [40]. In our research, however, we employed a k-nearest neighbor (k-NN) classifier with Euclidean distance in the recognition phase to focus only on the power of the feature extraction block.

The k-NN is a simple and fast supervised machine-learning algorithm which is used to classify the unlabeled testing set with a labeled training set. In order to classify a new instance, the system finds the k-nearest neighbors among the training dataset to the new input sample and uses the categories of the k-nearest neighbors to weight the category candidates. The prediction class of the testing input is found based on the minimum distance between the testing input data and the training samples [19].

The k-NN algorithm can be described by

$$Y(d_i) = \operatorname{argmax} \sum_{xj \in k\text{-}NN} \operatorname{Sim}(d_i, x_j)\, y(x_j, c_k), \quad (7)$$

where d_i is the testing sample, x_j is one of the neighbors in the training set, $\operatorname{Sim}(d_i, x_j)$ is the similarity function for d_i, and $y(x_j, c_k) \in \{0, 1\}$ indicates whether x_j belong to class c_k.

Finally, the class with maximal sum of similarity will be selected as testing sample class. The similarity function can be Euclidian distance, Mahalanobis distance, fringe distance, and so on.

Figure 8 shows the system's accuracy versus the number of features which are selected from an initial feature vector by the PCA. In general, accuracy peaks at a certain interval of features and then diminishes or saturates. In this experiment, the highest accuracies were achieved at intervals of 40 to 100 features.

To find the optimum number of features, the experiment was repeated with different number of features at intervals of 30 to 105 features using step 5. The best results were obtained for interval [75, 85]. The experiment was repeated for interval [75, 85] with step 1. Finally, the highest accuracy of 97.11% was achieved by using the first 79 features of the feature vector.

In the second experiment, and based on the results shown in Figure 8, the first 79 features proposed by PCA were selected as the final feature vector. In order to increase the recognition speed and also to evaluate the performance of the proposed sieving operation, we used different versions of the sieved training dataset. In all experiments, the number of features was 79. The results are shown in Table 1. This table clearly shows that the recognition speed increases to more than double, while the accuracy decreases slightly from 97.11% to 96.39%. If the reduced 1/3 version of the training dataset is used, the recognition speed will be more than 3 times faster, but the accuracy will drop by 3.03% from 97.11% to 94.08%.

4.3. Results Comparison. To compare the results in OCR domain, the results produced by the use of the proposed approach were compared with that from other available methods reported in the literature. Table 2 shows these comparisons. It is obvious that in most of the previous researches, MLP-NN was employed as the recognition engine. In order to have a better comparison, we repeated our experiments but used MLP-NN instead of k-NN at the recognition stage. At this stage, an MLP-NN was trained using 79 neurons in the input layer (corresponding to the number of selected features as in the previous experiment), 30 neurons in the hidden layer (found experimentally), and 10 neurons in the output layer (corresponding to 10 different classes of digits from 0 to 9), respectively. Each experiment was repeated 10 times and the obtained results have been reported in average. The achieved results have been shown in the last few rows in Table 2.

The results from the proposed method are better than the results for the other researches mentioned in Table 2, in terms of accuracy, except for the result obtained by Ziaratban et al. [31]. However, an accurate comparison between the proposed approach and other researches is not possible, because of the different databases used, number of training and testing samples, number of features, and the classifier employed.

To have a more accurate comparison with the recent related researches in Farsi OCR domain, we repeated our proposed approach under the same conditions in the method used by Enayatifar and Alirezanejad [32]. Both researches

Table 1: Recognition rate for different training dataset volumes.

Number of samples used for recognition operation by k-NN	Recognition time for a sample (seconds)	Ratio of recognition time to initial recognition time $T1$ (in percent)	Accuracy		
			$K = 1$	$K = 3$	$K = 5$
60,000	$T1 = 0.1161352$	$T1/T1 = 100\%$	97.11%	95.93%	95.70%
30,000	$T2 = 0.0509572$	$T2/T1 = 43.87748\%$	96.39%	94.97%	94.91%
20,000	$T3 = 0.0367815$	$T3/T1 = 31.67128\%$	94.08%	93.66%	93.61%
15,000	$T4 = 0.0296006$	$T4/T1 = 25.48805\%$	93.57%	93.40%	93.17%

Table 2: Result comparison for handwritten Farsi/Arabic digit recognition.

Year	References	Training dataset	Total number of testing samples	Number of features	Classifier	Accuracy
2003	Mowlaei and Faez [27]	(Private) (only 8 digits)	1600	64	SVM	92.44%
2003	Sadri et al. [28]	CENPARMI ver. 1	3035	64	MLP-NN	91.25%
2004	Mozaffari et al. [29]	(Private) (only 8 digits)	1600	64	MLP-NN	91.37%
2005	Mozaffari et al. [25]	(Private) (only 8 digits)	1600	240	Fr. NN	86.30%
2007	Ziaratban et al. [31]	(Private)	4000	60	MLP-NN	97.65%
2011	Enayatifar and Alirezanejad [32]	Hoda (60,000 samples)	20,000	48	MLP-NN	92.70%
2013	The proposed method	Hoda (60,000 samples)	20,000	79	k-NN	97.11%
2013	The proposed method	Hoda (30,000 samples)	20,000	79	k-NN	96.39%
2013	The proposed method	Hoda (20,000 samples)	20,000	79	k-NN	94.08%
2013	The proposed method	Hoda (15,000 samples)	20,000	79	k-NN	93.57%
2013	The proposed method	Hoda (60,000 samples)	20,000	79	MLP-NN	96.27%
2013	The proposed method	Hoda (30,000 samples)	20,000	79	MLP-NN	95.14%
2013	The proposed method	Hoda (20,000 samples)	20,000	79	MLP-NN	90.71%
2013	The proposed method	Hoda (10,000 samples)	20,000	79	MLP-NN	82.86%

have employed the same dataset, equal number of testing samples, and the same classifier (MLP-NN). However, the number of features in their study is 48, but we made a 79-dimensional feature vector. It is obvious that this value cannot decrease to 48. Also, our proposed approach introduces a new method for dataset reduction. Table 3 compares our proposed method results to the method proposed by Enayatifar and Alirezanejad in terms of accuracy and speed. Our proposed approach not only outperforms by achieving higher accuracy, but also it is about 4.5 times faster than the latter method.

4.4. Error Analysis. One of the main problems in Arabic/Farsi digit recognition using the template matching technique is that there is more than one general shape for digits 2, 3, 4, 5, and 6. Each of these digits is usually written in two (or more) different shapes in handwritten documents. Hence, the number of classes for Arabic/Farsi digits is 16 and not 10. These extra patterns degrade the templates generated for these digits. Hence, some templates are not similar enough to the sample images, and this causes the recognition system to produce the wrong results. To overcome this drawback, more than one template for the stated digits should be considered. Figure 9 shows the various shapes of the Arabic/Farsi digits 2, 3, 4, 5, and 6.

Another main source of errors in this research is the degraded samples in the testing part of the Hoda dataset, which cannot even be enhanced by the preprocessing block.

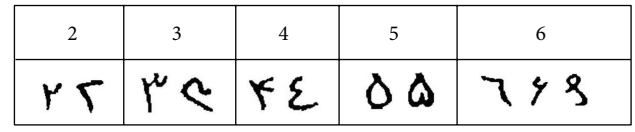


Figure 9: Different shapes for some of Arabic/Farsi digits.

Figure 10: Some degraded digit samples (first row) in the Hoda dataset.

It is also worth noting that there are usually some completely degraded samples as well as wrong samples in standard datasets (taken from real data) that are used to investigate the ability and behavior of a PR system to deal with these wrong or outlier samples. The Hoda dataset follows this rule. Figure 10 shows some degraded samples in the digit part of the Hoda dataset. Figure 11 shows the images of some degraded samples for digit 4 that had been misclassified as digits 2 or 3. These highly degraded samples have a negative impact not only on the template generating process but also on the recognition process.

Table 3: Result comparison for handwritten Farsi/Arabic digit recognition.

Year	References	Training dataset	Total number of testing samples	Number of features	Recognition time per sample (seconds)	Accuracy
2013	The proposed method	Hoda (40,000 samples)	20,000	79	0.036810	96.02%
2011	Enayatifar and Alirezanejad [32]	Hoda (40,000 samples)	20,000	48	(Not reported)	92.70%
2013	The proposed method	Hoda (7,000 samples)	3000	79	0.046235	95.72%
2011	Enayatifar and Alirezanejad [32]	Hoda (7,000 samples)	3000	48	0.21	94.30%

Table 4: Confusion matrix for the proposed method.

	۰ (0)	۱ (1)	۲ (2)	۳ (3)	۴ (4)	۵ (5)	۶ (6)	۷ (7)	۸ (8)	۹ (9)
۰ (0)	2000	0	0	0	0	0	0	0	0	0
۱ (1)	0	1998	2	0	0	0	0	0	0	0
۲ (2)	0	21	1928	36	5	2	2	6	0	0
۳ (3)	0	0	133	1823	37	2	0	3	2	0
۴ (4)	3	8	61	56	1856	2	7	3	0	4
۵ (5)	0	7	0	0	3	1966	3	2	14	5
۶ (6)	2	6	10	2	3	0	1952	5	2	18
۷ (7)	0	2	9	2	2	0	4	1979	0	2
۸ (8)	0	4	0	0	0	0	3	0	1987	6
۹ (9)	0	18	6	0	0	3	26	0	4	1943

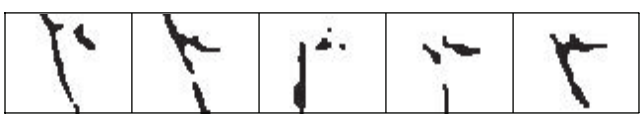


Figure 11: Some samples of digit 4 which were misclassified as digit 2 or digit 3.

As shown in the confusion matrix in Table 4, more than 57.74% of the errors are related to the misclassification of very similar digits 2, 3, and 4, in the handwriting mode. Most of the OCR systems for Arabic/Farsi language suffer from this too similar characteristic.

5. Conclusion

In this paper, a new dataset size reduction method was proposed in order to overcome the time complexity problem and to speed up the training and testing operations in an OCR application. To achieve this goal, we created a Modified Frequency Diagram, and we also developed a new method for creating a template for each class in the pattern space, while the similarity between each sample and its corresponding template was computed. This new similarity function was used to sort all the training data in each class based on the similarity to class template. After sorting all the data in each class, the training dataset size was sieved to 1/2, 1/3, and 1/4 of the original size by sampling at a rate of 1/2, 1/3, and 1/4.

In order to reduce the dimensionality, a PCA-based approach was introduced for automatic feature extraction and also for features selection in an OCR application for handwritten texts. Features were first extracted from the training patterns automatically by using the standard PCA. Then, using m biggest eigenvalues and their corresponding eigenvectors, a suboptimal feature set was selected from the initial feature vector. The input patterns were then mapped into the new orthogonal pattern space. A smaller dataset which has smaller number of samples and also smaller number of features was produced.

In the recognition stage, any input instance is mapped to a new pattern space using the computed suboptimal feature vector in the training phase. Classification is carried for an input instance by finding the nearest neighbor in the sieved dataset. The experiment was repeated using the MLP-NN instead of k-NN as the classification engine to compare the results with those reported in the literature.

The algorithm mentioned was implemented in an OCR system for recognizing digits in the Hoda dataset, which is one of the biggest standard handwritten Arabic/Farsi datasets. We achieved 97.11%, 96.39%, 94.08%, and 93.57% accuracy when all, 1/2, 1/3, and 1/4 training datasets, respectively, were used in the recognition phase using the k-NN classifier. In this case, the accuracy decreases from 97.11% to 96.39% while recognition speed increases by nearly two times. When a MLP-NN classifier was employed, the accuracies were 96.27%, 95.14%, 90.71%, and 82.86% corresponding to all, 1/2, 1/3, and 1/4 of training datasets, respectively. In

this case, the accuracy had decreased from 96.27% to 95.14%, while the recognition speed had increased by two times. Compared to one of the recent researches [32] in this domain, our proposed method achieved higher performance with recognition speed being about 4.5 times faster.

Identifying a systematic way to increase system recognition speed and introducing a new approach for dataset size reduction are two applications of the proposed method. The dataset size reduction algorithm is not only effective in OCR application—a subcategory of PR systems—but also can be used for dataset size reduction in other PR systems with different types of pictorial databases.

The current approaches used for dataset size reduction usually remove two groups of samples from the classes: near to classes' centroids and far from classes' centroids (outlier samples or support vector samples). However, the proposed method keeps some samples near to the classes' centers to make the system model better and also keeps some samples far from each class center to be able to assess system performance more accurately.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 99, pp. 1–24, 2013.

[2] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.

[3] H. B. Borges and J. C. Nievola, "Comparing the dimensionality reduction methods in gene expression databases," *Expert Systems with Applications*, vol. 39, pp. 10780–10795, 2012.

[4] M. Song, H. Yang, S. H. Siadat, and M. Pechenizkiy, "A comparative study of dimensionality reduction techniques to enhance trace clustering performances," *Expert Systems With Applications*, vol. 40, pp. 3722–3737, 2013.

[5] C. Yang, W. Zhang, J. Zou, S. Hu, and J. Qiu, "Feature selection in decision systems: a mean-variance approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 268063, 8 pages, 2013.

[6] G. A. Abandah, K. S. Younis, and M. Z. Khedher, "Handwritten Arabic character recognition using multiple classifiers based on letter form," in *Proceedings of the 5th IASTED International Conference on Signal Processing, Pattern Recognition & Applications (SPPRA '08)*, pp. 128–133, February 2008.

[7] W. Zhongdong, Y. Jianping, X. Weixin, and G. Xinbo, "Reduction of training datasets via fuzzy entropy for support vector machines," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 2381–2385, October 2004.

[8] S. V. N. Vishwanathan and M. N. Murty, "Use of multi category proximal SVM for data set reduction," *International Journal of Studies in Fuzziness and Soft Computing*, vol. 140, pp. 3–20, 2004.

[9] K. Hara and K. Nakayama, "Training data selection method for generalization by multilayer neural networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E81-A, no. 3, pp. 374–381, 1998.

[10] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, pp. 1–9, July 2004.

[11] S. Abdul Sattar and S. Shahl, "Character recognition of arabic script languages," in *Proceedings of the 2nd International Conference on Communication and Information Technology*, pp. 502–506, 2012.

[12] M. Elzobi, A. Al-Hamadi, L. Dinges, and B. Michaelis, "A structural features based segmentation for off-line handwritten Arabic text," in *Proceedings of the 5th International Symposium on I/V Communications and Mobile Networks (ISIVC '10)*, pp. 1–4, October 2010.

[13] H.-C. Kim, D. Kim, and S. Yang Bang, "A numeral character recognition using the PCA mixture model," *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 103–111, 2002.

[14] P. Zhang, C. Y. Suen, and T. D. Bui, "Multi-modal nonlinear feature reduction for the recognition of handwritten numerals," in *Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pp. 393–400, May 2004.

[15] V. Deepu, S. Madhvanath, and A. G. Ramakrishnan, "Principal component analysis for online handwritten character recognition," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 327–330, August 2004.

[16] A. Sharma and K. K. Paliwal, "Fast principal component analysis using fixed-point algorithm," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151–1155, 2007.

[17] M. T. Parvez and S. A. Mahmoudi, "Offline Arabic handwritten text recognition : a survey," *ACM Computing Survey*, vol. 45, no. 2, article 23, 2013.

[18] R. Azmi, B. Pishgoo, N. Norozi, M. Koohzadi, and F. Baesi, "A hybrid GA and SA algorithms for feature selection in recognition of hand-printed Farsi characters," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '10)*, vol. 3, pp. 384–387, October 2010.

[19] Y. El-Glaly and F. Quek, "Isolated Handwritten Arabic Character Recognition using Multilayer Perceptron and K Nearest Neighbor Classifiers," 2012, http://filebox.vt.edu/users/yasmineg/index_htm_files/MLSP%20Arabic%20Character%20Recognition.pdf.

[20] A. R. Kheyrkhah and E. Rahmanian, "Optimizing a Farsi handwritten character recognition system by selecting effective features on classifier using genetic algorithm," in *Proceedings of the 1st Joint Congress on Fuzzy and Intelligent Systems*, 2007 (Persian).

[21] A. M. Urmanov, A. A. Bougaev, and K. C. Gross, "Reducing the size of a training set for classification," US Patent no. 7478075 B2, 2007.

[22] I. Javed, M. N. Ayyaz, and W. Mehmood, "Efficient training data reduction for SVM based handwritten digits recognition," in *Proceedings of the International Conference on Electrical Engineering (ICEE '07)*, pp. 1–4, April 2007.

[23] B. Boucheham, "PLA data reduction for speeding up time series comparison," *International Arab Journal of Information Technology*, vol. 9, no. 5, 2012.

[24] J. Cervantes, X. Li, and W. Yu, "Support vector classification for large data sets by reducing training data with change of classes," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 2609–2614, October 2008.

[25] S. Mozaffari, K. Faez, and M. Ziaratban, "Character representation and recognition using quadtree-based fractal encoding scheme," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 819–823, September 2005.

[26] M. Ziaratban, K. Faez, and M. Ezoji, "Use of legal amount to confirm or correct the courtesy amount on Farsi bank checks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 1123–1127, September 2007.

[27] A. Mowlaei and K. Faez, "Recognition of isolated handwritten Persian/Arabic characters and numerals using support vector machines," in *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing*, pp. 547–554, 2003.

[28] J. Sadri, C. Y. Suen, and T. D. Bui, "Application of support vector machines of handwritten Arabic/Persian digits," in *Proceedings of the 2nd Iranian Conference on Machine Vision, Image Processing & Applications (MVIP '03)*, vol. 1, pp. 300–307, 2003.

[29] S. Mozaffari, K. Faez, and H. R. Kanan, "Feature comparison between fractal codes and wavelet transform in handwritten alphanumeric recognition using SVM classifier," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 331–334, August 2004.

[30] H. Soltanzadeh and M. Rahmati, "Recognition of Persian handwritten digits using image profiles of multiple orientations," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1569–1576, 2004.

[31] M. Ziaratban, K. Faez, and F. Faradji, "Language-based feature extraction using template-matching in Farsi/Arabic handwritten numeral recognition," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 297–301, September 2007.

[32] R. Enayatifar and M. Alirezanejad, "Offline handwriting digit recognition by using direction and accumulation of pixels," in *Proceedings of the International Conference on Computer and Software Modeling*, vol. 14, pp. 214–220, 2011.

[33] H. Khosravi and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1133–1141, 2007.

[34] M. Hanmandlu, K. R. Murali Mohan, S. Chakraborty, S. Goyal, and D. R. Choudhury, "Unconstrained handwritten character recognition based on fuzzy logic," *Pattern Recognition*, vol. 36, no. 3, pp. 603–623, 2003.

[35] A. C. Downton, E. Kabir, and D. Guillevic, "Syntactic and contextual post processing of handwritten addresses for optical character recognition," in *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 1072–1076, 1988.

[36] T. Sitamahalakshmi, A. Vinay Babu, and M. Jagadeesh, "Character recognition using Dempster-Shafer theory—combining different distance measurement methods," *International Journal of Engineering and Technology*, vol. 2, no. 5, pp. 1177–1184, 2010.

[37] V. Curic, J. Lindblad, N. Sladoje, H. Sarve, and G. Borgefors, "A new set distance and its application to shape registration," *Pattern Analysis and Applications*, vol. 12, pp. 1–12, 2012.

[38] A. Alaei, P. Nagabhushan, and U. Pal, "A new dataset of Persian handwritten documents and its segmentation," in *Proceedings of the 7th Iranian Conference on Machine Vision and Image Processing (MVIP '11)*, pp. 1–5, November 2011.

[39] M. Kherallah, A. Elbaati, H. E. Abed, and M. Alimi, "The on/off (LMCA) dual arabic handwriting database," in *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, 2008.

[40] Y.-C. Chim, A. A. Kassim, and Y. Ibrahim, "Dual classifier system for handprinted alphanumeric character recognition," *Pattern Analysis and Applications*, vol. 1, no. 3, pp. 155–162, 1998.