DATASET SIZE AND DIMENSIONALITY REDUCTION APPROACHES FOR HANDWRITTEN FARSI DIGITS AND CHARACTERS RECOGNITION

MOHAMMAD AMIN SHAYEGAN

THESIS SUBMITTED IN FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2015

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **MOHAMMAD AMIN SHAYEGAN** Passport No: **H95659872**

Registration/Matric No: **WHA100017**

Name of Degree: **DOCTOR OF PHILOSOPHY**

Title of Project Paper/Research Report/Dissertation/Thesis (“this Work”):

DATASET SIZE AND DIMENSIONALITY REDUCTION APPROACHES FOR HANDWRITTEN FARSI DIGITS AND CHARACTERS RECOGNITION

Field of Study: **DATA MINING**

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya (“UM”), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate’s Signature Date

Subscribed and solemnly declared before,

Witness’s Signature Date

Name:

Designation:

ABSTRACT

In all pattern recognition systems, increasing the recognition speed and improvement of the recognition accuracy are two important goals. However, these items usually perform against each other, when the former is improved, the latter is decreased, and vice versa. In this thesis, the focus is on both items; decreasing the overall processing time and increasing the system accuracy. To such an aim, the number of training samples is decreased by proposing a technique for dataset size reduction that leads to decrease of the training/testing time. Also, the number of features is decreased by proposing a new technique for dimensionality reduction. It decreases the training and testing time, and by deleting less important features, it increases the system accuracy, too.

The existing dataset size reduction algorithms, usually remove samples near to the centers of classes, or support vector samples between different classes. However, the former samples include valuable information about the class characteristics, and are important to make system model. The latter samples are important for evaluating system efficiency and adjustment of system parameters. The proposed dataset size reduction method employs Modified Frequency Diagram technique to create a template for each class. Then, a similarity value is calculated for each pattern. Thereafter, the samples in each class are rearranged based on their similarity values. Consequently, the number of training samples is reduced by **Sieving** technique. As a result, the training/testing time is decreased. In other part of this study, the number of extracted features is decreased by proposing a new method, which is, analyzing the one-dimensional and two-dimensional spectrum diagrams of standard deviation and minimum to maximum distributions for initial feature vector elements.

In recent years, the attractive nature of Optical Character Recognition (OCR) has caused the researchers to develop various algorithms for recognizing different alphabets. Target performance for an OCR system is to recognize at least five characters per second with 99.9% accuracy. However, the performance of available handwritten Farsi OCR systems is still_lacking, both in terms of accuracy and speed. The proposed techniques in this thesis have been validated in handwritten OCR domain via the use of two big standard benchmark datasets; the Hoda for Farsi digits and letters and the MNIST for Latin digits. The proposed dataset size reduction technique has been successful in decreasing the training time to less than half, while the accuracy has only decreased by 0.68%. Both datasets (Hoda and MNIST) were also used for dimensionality reduction purpose. Here, the dimension of feature vector was reduced to 59.40% for the MNIST dataset, 43.61% for digits part of the Hoda dataset, and 69.92% for the characters part of the Hoda dataset. Meanwhile the accuracies are enhanced 2.95%, 4.71%, and 1.92%, respectively. The achieved results showed the superiority of the proposed method compared to the rival dimension reduction methods.

The proposed size reduction technique can be used for other pictorial datasets. Also, the proposed dimensionality reduction technique can be employed in any other pattern recognition systems with numerical feature vectors.

ABSTRAK

Peningkatan kelajuan pengecaman dan ketepatan pengecaman adalah dua matlamat utama bagi kesemua sistem pengecaman corak. Walaubagaimanapun, kedua-dua faktor ini biasanya bertentangan antara satu sama lain di mana apabila prestasi kelajuan dipertingkatkan, prestasi ketepatan akan menurun dan begitu juga sebaliknya. Tesis ini memberi sasaran kepada kedua-dua faktor; pengurangan masa keseluruhan pemprosesan dan peningkatan prestasi ketepatan sistem. Untuk mencapai tujuan ini, satu teknik untuk mengurangkan saiz dataset bilangan sampel latihan telah dicadangkan yang membawa kepada pengurangan masa latihan/ujian. Satu teknik baru yang menyasarkan kepada pengurangan dimensi juga diperkenalkan supaya bilangan ciri-ciri turut berkurangan. Hasilnya, masa latihan dan ujian menjadi lebih pendek dan sistem juga menjadi lebih tepat dengan pembuangan ciri-ciri yang kurang penting.

Pada kebiasaannya, algoritma-algoritma sedia ada bagi mengurangkan saiz dataset akan membuang sampel-sampel yang berhampiran kepada pusat-pusat kelas atau menyokong sampel-sampel vektor di antara kelas-kelas berbeza. Tetapi, pusat-pusat kelas biasanya mengandungi maklumat penting berkenaan karakter-karakter atau ciri-ciri kelas tersebut yang penting untuk membina suatu model sistem. Vektor-vektor kelas pula penting untuk penilaian kecekapan dan pelarasan sistem. Kaedah pengurangan saiz dataset yang dicadangkan di dalam kerja ini mengambilkira teknik Gambarajah Pengubahsuaian Frekuensi untuk menjana satu templat untuk setiap kelas. Kemudian, satu nilai persamaan dikira untuk setiap corak. Selepas itu, sampel-sampel bagi setiap kelas disusun mengikut nilai-nilai persamaan tersebut. Ini mengakibatkan bilangan sampel-sampel latihan berkurangan dengan penggunaan teknik **Sieving**. Hasilnya, masa latihan/ujian menjadi pendek. Sebahagian lain dalam kerja ini telah mengkaji dan mencadangkan satu teknik baru untuk mengurangkan bilangan ciri-ciri yang diambil dengan menganalisa sisihan piawai serta pengagihan ciri asal elemen-elemen vektor daripada gambarajah-gambarajah spektrum satu-dimensi dan dua-dimensi.

Pada tahun-tahun kebelakangan ini, sifat menarik Pengecaman Huruf Optik (PHO) telah mendorong pengkaji-pengkaji untuk membina pelbagai algoritma-algoritma untuk mengenalpasti abjad-abjad yang berbeza. Prestasi sasaran untuk suatu sistem PHO adalah untuk mengecam sekurang-kurangnya lima huruf setiap saat dengan ketepatan 99.9%. Namun, prestasi bagi sistem-sistem PHO luar-talian untuk tulisan tangan Farsi masih jauh ketinggalan dari segi ketepatan dan kelajuan. Saiz dataset dan teknik-teknik pengurangan dimensi yang dicadangkan dalam tesis ini telahpun disahkan untuk domain tulisan tangan PHO dengan pengujian menggunakan dua dataset piawai yang terkenal; Hoda untuk huruf-huruf dan digit-digit Farsi dan juga MNIST untuk digit-digit Latin. Teknik pengurangan saiz dataset yang dicadangkan ini telah berjaya mengurangkan masa latihan kepada kurang daripada separuh dengan hanya menggunakan separuh daripada sampel-sampel latihan Hoda, manakala ketepatan telah meningkat sebanyak 0.68%. Kedua-dua dataset (Hoda dan MNIST) juga diuji untuk pengurangan dimensi. Di sini, dimensi-dimensi vektor ciri telah Berjaya dikurangkan kepada 59.40% untuk dataset MNIST, 43.61% untuk bahagian digit-digit daripada dataset Hoda, dan 69.92% untuk bahagian huruf-huruf dataset Hoda. Manakala ketepatan telah berjaya ditingkatkan sebanyak 2.95%, 4.71% dan 1.92% khasnya. Keputusan-keputusan ini amat menggalakkan dan membuktikan kelebihan mahupun keunggulan kaedah yang dicadangkan berbanding kaedah-kaedah pengurangan dimensi yang lain.

Teknik pengurangan saiz dataset yang dicadangkan juga boleh digunapakai oleh dataset-dataset berunsurkan gambar yang lain. Teknik yang dicadangkan juga boleh diaplikasi oleh

mana-mana sistem-sistem pengecaman corak yang berasaskan vektor-vektor bercirikan nombor.

ACKNOWLEDGEMENT

In the name of *ALLAH*, the most merciful and most compassionate. Praise to *ALLAH* who granted me strength, courage, patience and inspirations to complete this dissertation successfully.

This work would not have been possible without the nicest guidance from my research supervisor, Dr. Saeed Reza Aghabozorgi Sahaf Yazdi. He inspires me about the right way of the research.

I would also like to express my gratitude and thanks to my dear co-supervisors, Prof. Datin Dr. Sameem Abdul Kareem and Dr. Ram Gopal Raj, my previous supervisors Dr. Chan Chee Seng, and also Dr. Sandru Raviraja. Also, I would like to appreciate from Prof. Dr. Loo, Prof. Dr. Kabir, Dr. Alaee, and Dr. Khosravi for help, and advice throughout my graduate career. Without their support and guidance, none of the work presented in this thesis would have been possible.

To my lovely Family Fariba, Behnam, and Pedram for their patience and encouragement during my study.

Special appreciation to all my friends that help me to finish my study, mainly, Hassan Keshavarz, Afshin Sookhak, Alireza Tamjidi, Saeed Abolfazli, Azim Rezaei, and …

August, 2014

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

1D_MM	:	One-Dimensional Minimum to Maximum
1D_SD	:	One-Dimensional Standard Deviation
2D_MM	:	Two-Dimensional Minimum to Maximum
2D_SD	:	Two-Dimensional Standard Deviation
2S_SA		2 Stage - Spectrum Analysis
AI	:	Artificial Intelligence
BTM	:	Binarized Template Matrices
CBP	:	Connecting Broken Parts
COC	:	Change Of Classes
COM	:	Center Of Mass
CV	:	Cross Validation
DSA	:	Document Structure Analyses
FD	:	Frequency Diagram
FE	:	Feature Extraction
FHT	:	Farsi Handwritten Text
FOCR	:	Farsi Optical Character Recognition
FS	:	Feature Selection
GA	:	Genetic Algorithm
HMM	:	Hidden Markov Model
HT	:	Hough Transform
IFHCDB	:	Isolated Farsi Handwritten Character Data Base
IAUT/PHCN	:	Islamic Azad University of Tehran/Persian Handwritten City Names

k-NN	:	k-Nearest Neighbour
MFD	:	Modified Frequency Diagram
MNIST	:	Modified National Institute of Standards and Technology
MLP-NN	:	Multi-Layer Perceptron Neural Network
NN	:	Neural Network
OCR	:	Optical Character Recognition
PA	:	Partitioning Approach
PCA	:	Principal Component Analysis
PHTD	:	Persian Handwritten Text Dataset
PR	:	Pattern Recognition
PW	:	Pen Width
PWE	:	Pen Width Estimation
RP	:	Random Projection
SBR	:	Sieving Based Reduction
SBS	:	Sequential Backward Selection
SD	:	Standard Deviation
SI	:	Similarity Interval
SR	:	Similarity Ratio
ST	:	Sieving Technique
StaF	:	Statistical Features
StrF	:	Structural Features
SV	:	Similarity Value
SVD	:	Singular Value Decomposition
SVM	:	Support Vector Machine

TM	:	Template Matrices

CHAPTER 1

INTRODUCTION

1.1 Background

Pattern Recognition (PR) is one of the most important branches of Artificial Intelligence (AI) that correlates to observation and then classification. PR is concerned with designing and development of methods for the classification or description of objects, patterns, and signals. PR helps us to classify an unknown pattern using the previous knowledge or information driven from initial known patterns. Patterns are groups of observations or evaluations that define a set of points in a proper multidimensional space. A complete PR system is made of sensors, in order to receive the information that should be classified, methods for feature extraction and producing feature vectors, and pattern classification techniques reliance on extracted features in order to classifications.

Nowadays, great volume of available paper documents are converted to digital images documents by scanners, digital cameras, or even cell phones. Storing, restoring and efficient management of these images archives have great importance in many applications such as office automation systems, Internet-based documents searching, digital libraries, bank cheque processing, and zip code recognition (Parvez & Mahmoud, 2013). Consequently, achieving effective algorithm to analyze document image is an essential need.

The techniques that can recognize text zones in scanned images and then convert these zones to editable texts are called Optical Character Recognition (OCR) (Khosravi & Kabir, 2009). Machine simulation of human reading is another definition for OCR. An OCR system gets scanned images, recognizes its context (include: texts, lines, images, tables and so on) and then converts only the text parts to the machine-editable format. OCR systems increase several times the rate of data entry to the computer by deleting the typist role in converting the data from paper documents, in conventional media, into the electronic media format. Hence, demand for employing powerful OCR software is increasing rapidly. OCR systems can be used in many different applications such as: automated newspapers, automated mailing, automated banking, automated examining, implementation digital library, machine vision, and so on. The interesting nature of the OCR, as well as, its importance have created a lot of research orientation in different aspects and gained numerous advances (Jumari & A. Ali, 2002).

Without using OCR systems, access to information in the non-text documents is very difficult. Also, storing pictorial information need to very large memories. Hence, using OCR systems have two main advantages: **a)** More access to information, because there is a possibility to search and edit in texts against the images; **b)** Reducing storing spaces, because the volume of a text file is usually less than corresponding graphical file. These abilities prepare the possibility of wide spread of using computers for fast processing in different institutes like: banks, insurance companies, post offices and other organizations that face with millions of transactions, frequently (Ziaratban, Faez & Ezoji, 2007).

OCR systems are divided into two main groups, according to the manner in which input is provided to the recognition engine (Shah & Jethava, 2013): **i) Online systems:** In online systems, the patterns are recognized at the time of entering to the system. The input device for these systems is a digital tablet with a special light pen. In this method, in addition to information about the pen location, time information related to the pen path is used either.

This information usually is taken by a digitizer instrument. In this method, some information about speed, pressure and the time of putting and removing the pen on the digitizer are used; **ii) Offline systems:** In offline systems, the pre-saved input images of texts, obtained through the use of a scanner or a camera, are manipulated and recognized. In this method, there is no need to any kind of special editing tools, and the interpretation of input data is separated from the production process. This method is much similar to human style recognition (Khorsheed, 2002).

Online recognition is easier than offline recognition, because there are some important information, related to writing process in chronological order, such as speed and direction of the pen, order of writing strokes, the number of strokes, and relative location of complementary parts. Hence, online OCR systems are usually more accurate when compared to offline systems (Harouni, Mohamad & Rasouli, 2010; Ghods & kabir, 2013b, 2013c). However, based on their nature, the offline OCR systems are easier to apply on data, compared to online systems. Hence, most of the researches have been carried out on offline systems (Alaei, Nagabhushan & Pal, 2010a; Bahmani, Alamdar, Azmi & Haratizadeh, 2010; Jenabzade, Azmi, Pishgoo & Shirazi, 2011; Pourasad, Hassibi & Banaeyan, 2011; Rajabi, Nematbakhsh & Monadjemi, 2012; Ziaratban & Faez, 2012).

Another categorization method for OCR systems is related to type of entered data into the system (Lorigo & Govindaraju, 2006): **i) Printed:** If image data has been produced by machine (keyboards, type writers and so on), it is printed text; **ii) Handwritten:** If image data has been written by human, it is handwritten text.

Online OCR systems deal with only handwritten data, but offline OCR systems manipulate both of the printed and handwritten texts. In 1990's, recognition of printed patterns, including letters, digits and other popular symbols for different languages, has been studied by different groups of researchers. The results of these researches lead to find a collection of secure and fast OCR systems. First, researches are used to work on isolated letters and symbols, but now most of the research works are performed on connected letters (words and texts). Undoubtedly, OCR systems for printed texts are more established, while OCR systems for handwritten texts continue to attract more research efforts (Alginahi, 2012).

This chapter elaborates on the research motivation, problem statements, objectives, research methodology, contributions of the research, and organization of this thesis.

1.2 Research Motivation

The final goal of OCR systems is simulating human reading capabilities. They make interaction between man and machine in different applications such as automated banking, automated mailing, automated accounting, and so on. There are the huge databases on papers that if they are converted into machine form, then they can be used by other information systems. Also, in addition to the traditional OCR applications, there is a large interest in searching scanned documents which are available on the Web (Mahmoud & Mahmoud, 2006). Hence, OCR systems can help us to carry these demands very fast and accurate.

Most of the existing OCR systems have been designed for recognition of characters of Western languages with Roman alphabet, and also East Asian scripts such as Chinese and Japanese (Parvez & Mahmoud, 2013). Latin OCR business systems have had a considerably quality progress in recent years and the area has been considered as matured for recognition of printed non-cursive characters such as isolated Latin characters. However, there are distinct differences between Farsi and Latin letters, especially the cursive nature of Farsi alphabet in both printed and handwritten texts, and thus, it is not possible to use provided techniques for Latin text recognition directly for Farsi text recognition without first making some fundamental changes (Elzobi, Al-Kamdi, Dinges & Michaelis, 2010).

The Farsi alphabet had been derived from the Arabic alphabet, and it is the official language in Iran, Tajikistan and Afghanistan. About 30% of the world population and about 30 world languages use Farsi, Arabic, and similar alphabet as a base script for writing (Abdul Sattar & Shah, 2012) and this alphabet set is the second most-used alphabet set for writing worldwide (Elzobi, Al-Kamdi, Dinges & Michaelis, 2010). Hence, any advancement in OCR technology for Farsi alphabet set will bring widespread benefits.

Many researches have been carried out in the application of OCR technology for handwritten Arabic texts (Abandah & Anssari, 2009; Al-Hajj, Likforman & Mokbel , 2009; Al-Khateeb, Jiang, Ren, Khelifi & Ipson, 2009; Al-Khateeb, 2012; Bouchareb, Hamdi & Bedda, 2008; Sabri & Sunday, 2010; Dinges, Al-Hamadi, Elzobi, Al-Aghbari & Mustafa, 2011), and other languages which use Arabic-based alphabets. Most of these researches can also be adapted for Farsi letter recognition. However, there are a few important differences between the Farsi language and the Arabic language, such as the number of letters, different ligatures, different shapes for a letter in Farsi writing styles such as Nasta’’ligh or Shekasteh as compared to Arabic writing styles such as Naskh or Kufi. As a result, the Arabic OCR systems cannot be completely applied for Farsi documents. This is evident from the low recognition rate for handwritten Farsi texts, when using Arabic OCR products such as Sakhr Automatic Reader or ReadIris Pro for OCR of handwritten Farsi texts (Appendix I).

Research in Farsi OCR (FOCR) technology started in the early 1980’s by Parhami and Taraghi (1981), and this effort was followed by many research labs and universities across the globe with nearly acceptable results for printed Farsi documents or texts (Sadri, Izadi, Solimanpour, Suen & Bui, 2007; Kabir, 2009). Although more than one billion people worldwide use Farsi and other similar alphabets such as Arabic (Elzobi, Al-Kamdi, Dinges & Michaelis, 2010), Sindhi, Uygur, Kurdish, Sorani, Baluchi, Penjabi Shamukhi, Azer, Tajik (Abdul Sattar & Shah, 2012), Urdu (Khan & Haider, 2010), Jawi (Nasrudin, Omar, Zakaria & Yeun, 2008), Ottoman, Kashmiri, Adighe, Berber, Dargwa, Kazakh, Ingush, Kirghiz, Lahnda, Pashto (Zeki, 2005), and few others alphabets as their native language, but technical difficulties induced by the cursive nature of the Farsi documents have caused FOCR techniques have not been developed as perfectly as Latin, Japanese, Chinese, and even Arabic (Khosravi & Kabir, 2009) and system developed for identifying Farsi characters are not still efficient. Hence, online and offline recognition of these scripts have been in center of attention in the past few years.

Although there are some researches for recognition of handwritten Farsi digits with nearly acceptable results (Pan, Bui & Suen, 2009; Soltanzadeh & Rahmati, 2004), but available methods – due to of Farsi's alphabet special nature – are not completely extendable to Farsi alphabet characters. Also, the complex nature of cursive handwriting poses challenging problems in FOCR systems, and researches are still being carried out to find satisfactory solutions. Hence, available FOCR systems have a large distance from their real place, and research on this topic is still hot and demanding.

Some efforts have been made to develop OCR systems for handwritten Farsi characters, but the performance of these systems remains deficient in terms of accuracy and speed. The current FOCR systems for handwritten texts are far from achieving the target performance of recognizing five characters per second, with 99.9% accuracy with all errors being rejections (Khorsheed, 2002). Therefore, intense research efforts are needed to produce better FOCR systems.

1.3 Problems Statement

According to (Khorsheed, 2002), target performance for an OCR system is recognizing at least five characters per second with 99.9% accuracy. Hence, improvement the accuracy and increasing the recognition speed (decreasing the recognition time) are two main goals of any OCR systems. However, accuracy of conventional approaches for offline handwritten FOCR systems is not satisfactory enough. For example, the recognition rates of majority of available handwritten FOCR systems are in the range of 60% to 96% (Alaei, Nagabhushan & Pal, 2010a; Bahmani, Alamdar, Azmi & Haratizadeh, 2010; Enayatifar & Alirezanejad, 2011; Jenabzade, Azmi, Pishgoo & Shirazi, 2011; Pourasad, Hassibi & Banaeyan, 2011; Bahmani, Alamdar, Azmi & Haratizadeh, 2010; Salehpor & Behrad, 2010; Mozaffari, Faez, Margner and El-Abed, 2008b; Broumandnia, Shanbehzadeh & Varnoosfaderani, 2008; Gharoie Ahangar & Farajpoor Ahangar, 2009; Ziaratban, Faez & Allahveiradi, 2008).

To address this problem, the reasons for the low accuracy and high complexity in FOCR system were investigated. An OCR system has several different modules. The output of each module propagates to the next module in a pipeline fashion making the OCR system work as a whole and, if one stage fails, then the performance is significantly affected. In a FOCR system, all modules ‘data acquisition’, ‘pre-processing’, ‘segmentation’, ‘feature extraction’, and ‘recognition’ are important. However, the majority of these parts are not in satisfactory conditions for handwritten FOCR systems. For example, low quality of pre-processing block output, large number of less-important training samples, heuristic methods for feature extraction, and a large number of non/less important features are among these weaknesses. Hence, to support the problem statement, that the performance of FOCR systems should be improved, the following sub-problems are explained:

i) The existence of deficiencies in output of pre-processing block: There are some powerful pre-processing techniques, such as noise removal, normalization, smoothing, de-slanting, and so on, which cause very good results in OCR systems (Table 4.1). However, there are still some weaknesses in the output generated by pre-processing block. For example, there are some dis-connected parts in scanned image of Farsi or English digits, based on different reasons such as low quality of employed scanners, low quality of initial images, low quality of papers, and so on. The current pre-processing methods cannot attach these broken parts of an image together, and reconstruct the initial image. These degraded samples will cause a noticeable negative impact on recognition accuracy. Hence, it is necessary to find more efficient algorithms for this task.

ii) Large number of less-important training samples: In all PR systems, the quantity, quality, and diversity of training data in the learning process directly affect the final results. In this context, the size of the training dataset is a crucial factor, because the training phase, for making the system model, is often a time-consuming process. Also, the required computational time for classifying input data increases linearly (such as in *k*-NN) or nonlinearly (such as in SVMs) with the number of samples in the training dataset (Urmanov, Bougaev & Gross, 2007). For example, time complexity for SVM classifiers grows with the square of the number of samples in the training dataset (Zhang, Suen & Bui, 2004). Hence, some of powerful classifiers cannot be used in online or offline PR applications with very large number of training samples. As an example, the mentioned powerful classifiers maybe cannot be used in license plate recognition application, in real situations.

Nowadays, dataset sizes have grown dramatically (Kuri-Morales & Rogriguez-Erazo, 2009), but a major problem of PR systems is due to the large volume of training datasets including duplicate and similar training samples. Usually, the similar and repetitive samples not only do not feed different valuable information into a PR system, but also increase training time (and sometimes testing time) of the system. These similar samples need to a large memory for storing, too. In addition, there is an increasing demand for employing various applications on limited-speed and limited-memory devices such as mobile phones and mobile scanners (Sanaei, Abolfazli, Gani and Buyya, 2013). Therefore, it would be very beneficial to be able to train a PR system with a smaller version of training datasets, without incurring significant loss in system accuracy. Reducing the volume of initial data is an important goal toward speeding up the training and testing processes. 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.

iii) High dimensionality of feature space, because of existence of less-important features: Feature selection is another important step in PR systems. Although there are different conventional approaches for feature selection, such as Principal Component Analysis (PCA), Random Projection (RP), and Linear Discriminant Analysis (LDA), selecting optimal, effective, and robust features, in OCR applications, is usually a difficult task **(**Abandah, Younis & Khedher, 2008**)**. Therefore, it is still necessary to find new methods compared to conventional methods.

Finally, the problem of this study is stated briefly as: "**Performance of available offline handwritten FOCR systems is still far from humans, both in terms of accuracy and speed, because of large number of training samples, high dimensionality of feature space, existence of some deficiencies in output of pre-processing block, and lack of optimal, effective, and robust features set.**"

1.4 Research Questions

The research questions which are answered through this study are as follows:

- Q1. How to enhance the output of the pre-processing step in OCR systems?

- Q2. What is the impact of dataset size reduction on the accuracy of handwritten FOCR systems?

- Q3. How to reduce dimensionality of feature space and increase the recognition accuracy, simultaneously, in FOCR system?

- Q4. What is the best features set for handwritten FOCR systems?

1.5 Research Aims and Objectives

The main goal of this research are increasing the recognition accuracy in offline handwritten FOCR systems and also increasing the recognition speed. However, there is usually a tradeoff between the accuracy and recognition speed, i.e. enhancement in one of these two parameters usually means defect in another one. Hence, the main objectives are fourfold as follows:

1) To enhance the output quality of pre-processing block in a handwritten FOCR system: In order to increase recognition accuracy, a new approach, for connecting the broken parts of an image together, is proposed to enhance the quality of pre-processing output.

2) To propose a new technique for dataset size reduction, in a handwritten FOCR system, to speed up system training and testing: In order to save processing time and memory usage in training part of an FOCR system, a new method for dataset size reduction, without significant negative effect on final accuracy, is proposed. Reducing the number of training samples not only decreases the overall training time, but also it decreases the testing time, in the case of using special classifiers (such as *k*-NN).

3) To propose a new technique for dimensionality reduction, in a handwritten FOCR system, to speed up system training and testing, and increasing the system accuracy: Similar to dataset size reduction, dimensionality reduction (feature selection, feature reduction) can save the processing time and memory usage in training and testing steps of an OCR system. Hence, finding a new and efficient method for dimensionality reduction is another objective of this research. It leads to introduce small features set for handwritten FOCR systems.

4) To test and evaluate the capability of the proposed methods in improving the performance of a FOCR application, by applying them on Farsi digits and characters: All the proposed methods and techniques are validated by using standard benchmark OCR dataset Hoda, and in some cases by using standard benchmark OCR dataset MNIST.

In brief, improving the quality of pre-processing step results, finding a new algorithm for dataset size reduction (in order to reduce processing time), and finding a new and efficient algorithm for dimensionality reduction (in order to reduce processing time and increase the final accuracy), are the final goals in this thesis.

1.6 Research Scope and Limitations

Handwritten characters recognition is considered as one of the most challenging and exciting areas of research in PR domain. This is partly due to the diversity of sizes, fonts, orientation, shapes, thickness, and dimension of characters in handwritten texts resulting from different writing habits, styles, educational level, moods, health status and other conditions of the writers. In addition, other factors such as the writing instruments, writing surfaces and scanning methods, along with other problems, such as unwanted characters overlapping in sentences in any language, make handwritten scripts recognition much more difficult than recognition of printed texts (Ghods & kabir, 2013a). As a result, OCR systems for handwritten texts do not perform as well as OCR systems for printed texts (Mandal & Manna, 2011; Fouladi, Araabi & Kabir, 2013).

To ensure that this research can achieve its set of objectives, within the stipulated timeframe, some limitations on type of input data, vocabularies, writing style, and so on in handwritten OCR systems, need to be defined:

1) Manipulating one of available free standard Handwritten Farsi datasets.

2) Manipulating alone mode of Farsi letters. To process all modes of Farsi letters, i.e. beginning mode, middle mode, end sticky mode, and alone mode, need to apply external segmentation process on handwritten Farsi words, and external segmentation is not in this study scope.

3) Implementing only related parts to objectives, not for a whole OCR system.

1.7 Research Methodology

This research deals with offline handwritten Farsi character recognition. Research methodology and systematic way concerned to this work can be expressed as follows:

Firstly, the problems related to offline handwritten Farsi character recognition have been investigated by library research (by reading and reviewing the previous published researches for FOCR in proceedings and journals). Also, the different available approaches for this topic are reviewed. Secondly, the limitation and scope of existing approach are pointed. Thereafter, a model for a FOCR system is proposed including new modules Connecting Broken Parts, Dataset Size Reduction, and Dimensionality Reduction (Features Selection).

For evaluating the overall system performance, it should be measured the accuracy and speed of the proposed model. Hence, an appropriate standard benchmark FOCR dataset is selected to test the proposed algorithms, and the system performance is computed. Also, in some experiments the k-fold cross validation is used. In addition, to evaluate the efficiency of the proposed methods for non-Farsi datasets, the English benchmark standard dataset MNIST, is employed, too. Finally, the outcome results will be compared to the most related literatures in the almost same conditions. Our approach for this research work is implementation driven and experimentally. Also, appropriate tools such as suitable implementation language (C#, MATLAB, …) are chosen.

1.8 Organization of the Thesis

This thesis consists of six chapters. The current chapter introduces some key definitions in OCR domain including: FOCR, printed and handwritten texts, online and offline recognition; addressing the problems statement; identifying research significance and limitations; and also research objectives.

Chapter 2 introduces in detail literatures on different parts of a FOCR system, including: data acquisition, pre-processing, features extraction/selection, and also recognition. Also, the literature about dataset size and dimensionality reduction, which are considered in this thesis, is introduced. However, the main direction of all subjects is around FOCR systems.

The used methodology and the proposed model for recognizing handwritten Farsi digits and letters are covered in Chapter 3 to achieve research objectives.

Chapter 4 demonstrates the main part of the thesis, which is the presentation and design of the proposed FOCR model. A new proposed method to connect broken parts of an image together (in pre-processing block), a new proposed method for dataset size reduction using Modified Frequency Diagram Matching (in system training phase) and new similarity measurement function, and finally, a new proposed dimensionality reduction technique by employing one and two dimensional standard deviation and minimum to maximum spectrum diagrams analysis (in system training and testing phases) are explained, completely. Also, all the applied operations on the input data including pre-processing, size reduction, feature extraction and selection, and recognition are explained, in detail. Many aspects of this thesis are discussed in this chapter, including the methods designed to achieve the study's objectives. It also reports the final results from experimental data used in this thesis.

Chapter 5 evaluates and discusses the proposed model from different aspects. It includes evaluation and results comparison of the proposed size and dimensionality reduction techniques with the most related literature, explaining the advantages and disadvantages of the proposed methods, and also analyzing the occurred errors in recognition step.

Finally, the thesis concludes with Chapter 6, which introduces conclusions and discussion how the research objectives were met. The contribution of the research is pointed again, followed by some guidelines and suggestions for future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Document image analysis covers the algorithms which they transform documents into electronic format suitable for storage, retrieval, search, and update process. Optical Character Recognition (OCR) systems convert graphical images into editable texts. The OCR technology is now widely used, and research and development on its applications is on-going (Parvez & Mahmoud, 2013).

2.1.1 Printed / Handwritten Text

OCR systems are categorized into two main groups, based on the type of input data entered into the systems: "Printed" or "Handwritten". In printed mode, various fonts of machines, computer keyboards, printers, and so on are considered as input data. In this mode, the inputs usually have a good quality, because the machines generate them. Hence, the recognition process is simpler than the second group, and efficiency of the system are usually noticeable. These systems are generally used for recognizing printed documents such as books, newspapers, and other similar documents.

In contrast, handwritten documents are produced by different people in different situations. Hence, handwritten characters recognition is considered as one of the most challenging and exciting areas of research in Pattern Recognition (PR) domain. This is partly due to the diversity of sizes, orientation, thickness, and dimension of characters in handwritten texts resulting from different writing habits, styles, educational level, moods, health status and other conditions of the writers. In addition, other items such as the writing instruments, writing surfaces and scanning methods, along with other problems such as unwanted characters overlapping in sentences, in any language, make handwritten scripts recognition very difficult.

Based on obeying some limitations and rules, handwritten texts are divided into two sub-categories: constrained and unconstrained. There are some regularities in constrained writing style. By this reason, recognition operation for this type of texts is faster, easier, and more accurate in comparison with unconstrained texts. However, the general shapes of characters in this case are not similar to real situation. For example, character dimensions, character slants, and so on have been predefined.

Recognition of handwritten texts is much more difficult than recognition of printed texts (Ghods & Kabir, 2013a). Different writing styles lead to the distortion in input patterns from the standard patterns (Mandal & Manna, 2011). Therefore, unlike printed OCR systems, that they have been matured, handwritten OCR systems are still open research area and there is a long way to their final goals. Consequently, OCR systems for handwritten texts do not perform as well as OCR systems for printed texts (Mandal & Manna, 2011; Fouladi, Arrabi & Kabir, 2013). Undoubtedly, OCR systems for printed texts are more established, while OCR systems for handwritten texts continue to attract more research efforts (Alginahi, 2012).

2.1.2 Online / Offline OCR Systems

Generally, OCR systems are divided into two main groups "Offline" and "Online", based on when the recognition operation is carried out. In offline method, recognition operation is performed after the writing or printing process is completed, but in online systems, recognition is carried out in the same time of entering the data to the system (Shah & Jethava, 2013).

In online OCR systems, information is imported into system by using digitized tablets and a stylus pen. In every moment, x and y coordinates of the pen tip on the page, the value of pen pressure on the page, angle and direction of writing and so on, are useful information for this group of systems. In this case, there are some important information related to writing characters in chronological order such as order of writing strokes, number of strokes, speed and direction of pen, and location of complementary parts related to main parts of a character. Hence, online OCR systems are usually more accurate when compared to offline systems (Baghshah, Shouraki & Kasaei, 2005, 2006; Faradji, Faez & Nosrati, 2007; Faradji, Faez & Mousavi, 2007; Halavati & Shouraki, 2007; Harouni, Mohamad & Rasouli, 2010; Samimi, Khademi, Nikookar & Farahani, 2010; Nourouzian, Mezghani, Mitichi & Jonston, 2006; Ghods & Kabir, 2010, 2013b, 2013c).

In offline systems, both type of printed or handwritten texts are converted to graphical files by special devices such as scanner, digital cameras, or even cell phones, and then imported to an OCR system. In this type of OCR systems, recognition operations are performed after writing process. Hence, no auxiliary information associated with images are available to the system. Offline recognition of handwritten cursive text (such as Farsi text) is very more difficult than online recognition, because the formers must deal with 2D images of the text, after it has already been written (Lorigo & Govindaraju, 2006). Offline recognition of unconstrained handwritten cursive text must overcome many difficulties such as similarities of distinct letter shapes, unlimited variation in writing style, characters overlapping and interconnection of neighboring letters. However, based on their nature, the offline OCR systems are easier to apply than the online systems. Hence, most of the researches have been carried out on offline systems, and this is also true for the Farsi OCR (FOCR) systems (Abedi, Faez, & Mozaffari, 2009; Alaei, Nagabhushan & Pal, 2010a; Bahmani, Alamdar, Azmi & Haratizadeh, 2010; Enayatifar & Alirezanejad, 2011; Jenabzade, Azmi, Pishgoo & Shirazi, 2011; Pourasad, Hassibi & Banaeyan, 2011; Rajabi, Nematbakhsh & Monadjemi, 2012; Salehpor & Behrad, 2010; Ziaratban & Faez, 2012).

The available useful information in online recognition systems have caused researchers try to extract some of these information for offline systems, too. They try to develop some approaches to find distribution of the image pixels (identical to online methods) from available information in offline handwriting texts. For an example, Elbaati, Kherallah, Ennaji and Alimi (2009) tried to find strokes temporal order from a scanned handwritten Arabic text for using them in an offline Arabic OCR system. They extracted some features such as end stroke points, branching points, and crossing points from the image skeleton. After that, they tried to find the order of strings in each stroke. They used also genetic algorithm for finding the best combination of stroke order.

2.2 Farsi Writing Characteristics

Handwritten Farsi documents have unique characteristics based on cursive orthography and letter shape context sensitivity. There are a few characteristics and features which make Farsi cursive writing unique when compared to other languages. They cause that innovated methods for recognition of other languages are not exactly suitable for Farsi by the same conditions.

In this section, some of the main characteristics of Farsi scripts will be briefly described to point out the main difficulties which an FOCR system should overcome.

- **Farsi Alphabet**: Farsi alphabet involves 32 basic letters. Figure 2.1 shows a sample of the whole handwritten isolated mode of Farsi letters.

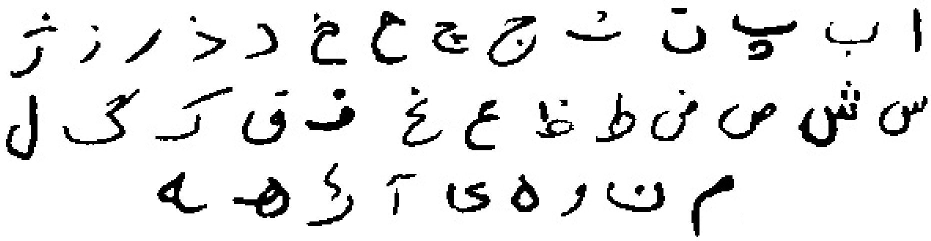


Figure 2.1 : A sample of isolated mode of Farsi letters

- **Writing Direction**: Farsi texts are written from right to left direction on an (or more) imaginary horizontal line(s) called baseline(s), as compared to Latin, but numeral strings are written from left to right similar Latin.

- **Cursive language**: By nature, Farsi writing is cursive, even in machine-printed forms, which means letters stick together from one or two sides to make the sub-words. However, some letters are written separately. The cursive nature of the Farsi texts is the main obstacle to any FOCR system. For handling this situation, sometimes FOCR systems need to use external segmentation operation to disjoint connected letters. However, segmentation is one of the bottlenecks steps in FOCR systems. This subject causes the performance of FOCR systems is lower than of Latin OCR systems.

- **Sub-words:** seven out of 32 Farsi letters (ا , د , ذ , ر , ز , ژ , و) cannot be linked by the left succeeding letter in a word and they stick only to previous letter. Therefore, if one of these letters exits in a word, it divides the word into two or more sub-words.

- **Different shapes for letters**: Farsi letters shapes are content sensitive according to their location within a word, where each letter can take up to four different shapes, as shown in Table 2.1. These forms are: Beginning (or Initial), Middle, End Sticky and Isolated (or Alone). This fact has caused that although the number of Farsi alphabet is 32, but they appear in more than 120 different shapes.

Table 2.1 : Farsi alphabet and their different shapes

Farsi Letters	Initial Mode **I**	Middle Mode **M**	End Sticky Mode **E**	Alone Mode **A**
ا	----	----	ﺎ	ا
ب	ﺑ	ﺒ	ﺐ	ب
پ	ﭘ	ﭙ	ﭗ	پ
ت	ﺗ	ﺘ	ﺖ	ت
ث	ﺛ	ﺜ	ﺚ	ث
ج	ﺟ	ﺠ	ﺞ	ج
چ	ﭼ	ﭽ	ﭻ	چ
ح	ﺣ	ﺤ	ﺢ	ح
خ	ﺧ	ﺨ	ﺦ	خ
د	----	----	ﺪ	د
ذ	----	----	ﺬ	ذ
ر	----	----	ﺮ	ر
ز	----	----	ﺰ	ز
ژ	----	----	ﮋ	ژ
س	ﺳ	ﺴ	ﺲ	س
ش	ﺷ	ﺸ	ﺶ	ش
ص	ﺻ	ﺼ	ﺺ	ص
ض	ﺿ	ﻀ	ﺾ	ض
ط	ﻃ	ﻄ	ﻂ	ط
ظ	ﻇ	ﻈ	ﻆ	ظ
ع	ﻋ	ﻌ	ﻊ	ع
غ	ﻏ	ﻐ	ﻎ	غ
ف	ﻓ	ﻔ	ﻒ	ف
ق	ﻗ	ﻘ	ﻖ	ق
ک	ﮐ	ﮑ	ﮏ	ک
گ	ﮔ	ﮕ	ﮓ	گ
ل	ﻟ	ﻠ	ﻞ	ل
م	ﻣ	ﻤ	ﻢ	م
ن	ﻧ	ﻨ	ﻦ	ن
و	----	----	ﻮ	و
ه	ﻫ	ﻬ	ﻪ	ه
ی	ﯾ	ﯿ	ﻲ	ی

- **Similar characters**: More than half the Farsi letters share the same main body. They are differentiated in terms of the number and location of some complementary (secondary) parts such as dots, zigzags, slanted bars, and so on. The similarity of the letters can cause further problems with classification, when noise is added to these similar characters. Table 2.2 shows the different groups of Farsi character which share similar bodies.

Table 2.2 : Different groups of Farsi letters and digits with similar bodies

Groups	**Similar Characters**
1	ب ، پ ، ت ، ث
2	ج ، چ ، ح ، خ
3	د ، ذ
4	ر ، ز ، ژ
5	س ، ش
6	ص ، ض
7	ط ، ظ
8	ع ، غ
9	ف ، ق
10	ک ، گ
11	۲ ، ۳ ، ۴
12	۱ ، ٦ ، ۹

- **Dots**: Eighteen out of 32 Farsi letters (more than 56%) have one, two or three dot(s) above, below or in the middle of letters body. It is worth noting that any erosion or removal of these dots will lead to a misrepresentation of the letters. Therefore, efficient pre-processing techniques have to be used in order to keep these dots and avoid misunderstanding during processing such as noise removal for image enhancement (Al-Khateeb, Jiang, Ren, Khelifi & Ipson, 2009).

- **Dots Shapes**: Styles of writing (shape and size) of dots are different from person to person in handwritten documents. Therefore, sometimes it is necessary to consider extra classes for these new shapes of dots. Figure 2.2 shows some examples of writing 3-dots pattern.

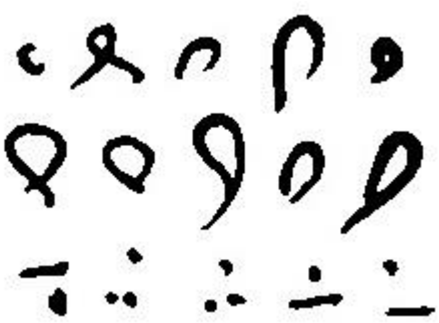


Figure 2.2 : Some style of writing 3-Dots in Farsi letters and words

- **Secondary parts:** Some Farsi letters have extra parts such as slanted bars, "Hamzeh", "Tanvin", and "Tashdid" symbols. The majority of hasty writers draw the secondary components in wrong position or even they attach them to the main letter body. It sometimes causes a lot of difficulty in finding and recognition these secondary parts.

- **Baseline**: Farsi characters are lied down on imaginary horizontal lines (called baselines), where letter connections are occurred and from where descending and ascending letters extend.

- **Jags**: A considerable percentage of Farsi letters (especially sticky letters) have jags near to baselines. If the original documents have low quality, or scanner has low resolution, then these jags are appeared in very small size and they are not seen by the system. This subject produces several errors in segmentation and recognition phases.

- **Ligature:** In both printed and handwritten mode of writing, two or occasionally three letters can be combined vertically, in an accepted manner, to form a new unit shape, called ‘ligature’ (Lorigo & Govindaraju, 2006). Ligatures are exceptions from the joining letters’ rules to make a sub-word. One example is combining letters ‘ ﻟ ‘ and ‘ ﺎ ‘ and producing ligature ‘ ﻻ ‘. The most vertical ligatures are not obligatory, and they are appeared for the aesthetic reasons (Sari & Sellami, 2007). Usually, segmenting a ligature to initial letters is very difficult. Therefore a ligature is considered as a new pattern in a dataset. It causes the increasing in the number of patterns in pattern space.

- **Vertical Overlapping**: Majority of neighbor letters in handwritten Farsi words may overlap vertically, without any touching. Hence, these letters cannot separate completely from each other by drawing a simple vertical line. In addition, extra parts of letters, such as slanted bar, usually overlap the adjacent letters in a word.

- **Different Dimensions:** The height and width of Farsi characters vary across various characters and across the different shapes of the same character in different position in a word, even in printed form (Table 2.1). For example letter ‘ ک ‘ and letter ‘ ه ‘ have not equal height and width.

- **Intra Space:** The space between two sub-words does not have a standard amount in handwritten Farsi texts.

- **Confusing Characters:** Some Farsi letters are very similar to digits, such as: letter ‘ ا ‘ and digit ‘ ۱ ‘ , letter ‘ ه ‘ and digit ‘ ۵ ‘, and letter ‘ . ‘ and digit ‘ ۰ ‘. This characteristic leads the recognition module to error.

- **Extra forms for a character:** Some Farsi digits have more than one form, such as: digits ‘ ٤ ‘ and ‘ ۴ ‘, digits ‘ ٢ ‘ and ‘ r ‘, digits ‘ ٥ ‘ and digit ‘ ۵ ‘, digits ‘ ٦ ‘ and ‘ ۶ ‘.

- **Sloping and multiple baselines:** Some of Farsi writing styles such as Nasta’aligh, have more than one baseline in each line of a text and these baselines are not horizontal in nature.

- Many of Farsi letters have ascending and descending part which are salient characteristics for recognition.

Table 2.3 and Figure 2.3 show the mentioned characteristics.

Table 2.3: Some Farsi letters and their characteristics

Transcriptions	Farsi Alphabets	Initial Mode	Middle Mode	End Sticky Mode	Alone Mode	Characteristics
Dal	د	----	----	ﺪ	د	A letter with 2 different forms
Be	ب	ﺑ	ﺒ	ﺐ	ب	A letter with 4 different forms
Jeem	ج	ﺟ	ﺠ	ﺞ	ج	11: Different letters with similar bodies
Chaa	چ	ﭼ	ﭽ	ﭻ	چ	
Ghain	غ	ﻏ	ﻐ	ﻎ	غ	12: Different shapes for a letter
Raa	ر	----	----	ﺮ	ر	13: A letter without dot
Zal	ذ	----	----	ﺬ	ذ	13: A letter with one dot
Qaaf	ق	ﻗ	ﻘ	ﻖ	ق	13: A letter with two dots
Shin	ش	ﺷ	ﺸ	ﺶ	ش	13: A letter with three dots
Gaaf	گ	ﮔ	ﮕ	ﮓ	گ	14: Letters with different dimensions; 7: Slanted bar
Meem	م	ﻣ	ﻤ	ﻢ	م	

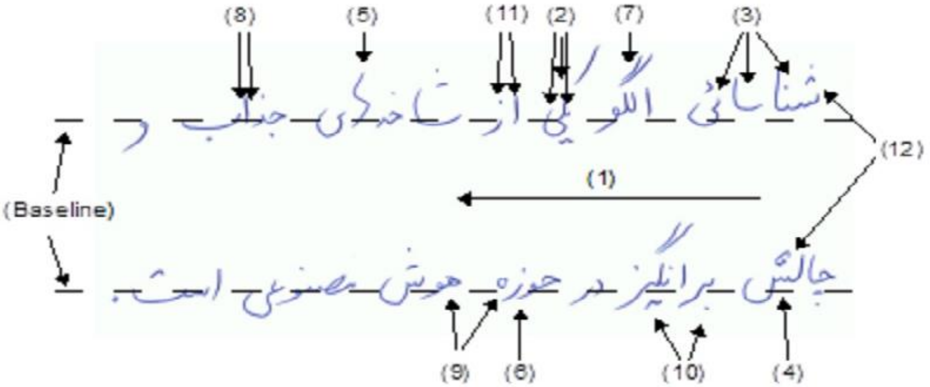


Figure 2.3: Some various aspects of Farsi writing characteristics.

(1): Writing direction, (2): Creating a word using sticky letters (3): Creating a sub-word using non-sticky letters in the middle of a word, (4): Jags, (5): Ligature, (6): Vertically overlapping, (7): Extra part of a letter, (8): Various types of spaces between sub-words, (9): Different shape of a letter, (10): Dots, (11): Various heights and widths for different characters, (12): Various shapes and sizes of dots.

Although most of the techniques used in FOCR systems are not fundamentally different from those used in Latin OCR systems, but there are some special linguistic rules associated with Farsi writing that render Farsi character recognition task more challenging than that for Latin. The aforementioned characteristics have prompted researchers to examine some of the problems encountered, which have only recently been addressed by researchers of other languages. These problems are the main obstacles of developing OCR systems for Farsi language (and similar alphabet languages) (Khorsheed, 2002). Fortunately, there are evidences of intense efforts being made to overcome the FOCR problems.

2.3 Different Modules in a FOCR System

Reviewing the existing research works in literature shows that basically an FOCR system has the following components: Data Acquisition, Pre-processing, Segmentation, Feature extraction, and Recognition. Figure 2.4 depicts an overview of these components.

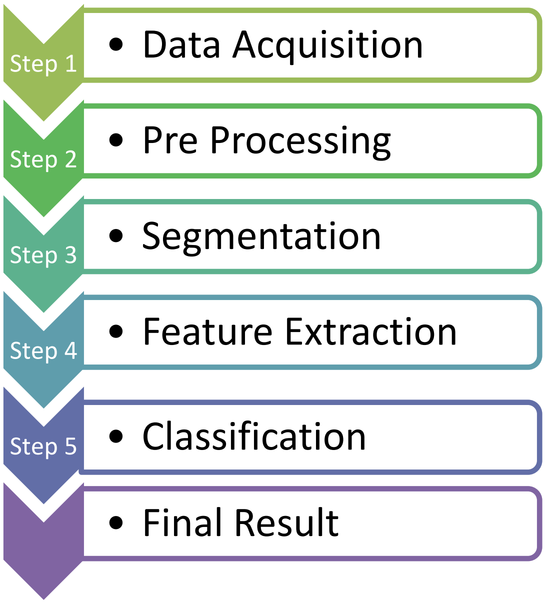


Figure 2.4: Available blocks in an FOCR system

Among the available blocks in Figure 2.4, the important ‘Segmentation’ block does not considered in this research, because it is out of scope of this study. However, a literature about segmentation process in FOCR systems reported in Appendix II. The other parts are demonstrated in the following sub-sections.

2.3.1 Image Acquisition and Available Farsi OCR Datasets

2.3.1.1 Introduction

Any PR system needs to use appropriate initial data (or standard datasets) for training and also for testing the efficiency and effectiveness related algorithms. In other word, datasets are one of the basic requirements in all studies in PR domain. Appropriate datasets allow researchers to test the efficiency of their proposed algorithms in real applications. An OCR system, as a sub-category of a PR system, must follow this rule and therefore, all OCR systems need to have appropriate datasets. However, producing or gathering data, to generate a comprehensive dataset, is a difficult, costly and time-consuming activity.

In the beginning, the researchers used to collect (or produce) datasets to test the OCR system performance designed by their own. Hence, most of the developments in OCR research have been benchmarked on private datasets. However, in many cases, these collections are not complete and therefore, they were not suitable criteria for evaluation of different systems. For example, it is possible that a weak PR system, with using a few high quality numbers of samples, achieves to high accuracy, while another PR system, despite being more complete, but due to the use of comprehensive and realistic data, shows less accuracy in recognition. In other word, the second system will gained lower acceptance despite the greater merit. It leads to mistaken judgments about these two PR systems. As a result, different methods are not comparable to one other, when they are tested on different datasets. Therefore, employing the benchmark standard datasets allow a fair comparison between different methods and systems. Some general features related to standard OCR datasets are (Kabir, 2009):

- To be pervasive enough. It means a dataset should include all of the possible patterns (letters, digits, common symbols and varieties of each form of them).
- Existence of adequate number of each pattern, taken from real environments.
- Variation in data collection devices.
- Diversity in the primary documents writers.
- Unconstrained writing process.
- Independency to datasets preparation methods.
- Upgrading ability for updating the information.
- Using various sources for gathering information as possible.
- According to end users demands.
- Easy to use (not include unnecessary overheads)
- Availability (low cost and even free)
- Including the suitable categories (every user can use data according to his or her needs and requirements, and do not need to store unnecessary large volume of data).
- Use popular media data transfer (particularly the Internet).
- Existing equivalent information required datasets images (Ground Truth information, metadata information).
- …

Usually, a dataset is divided to two different parts: ‘Training Set’ and ‘Testing Set’. Sometimes, in order to reduce some of the errors related to system training, a part of the data is categorized to the third group ‘Validation Set (Verifying Set)’. In this case, classifier parameters are adjusted by testing the designed system on validation set for increasing the accuracy of the system. However, existence of two training and testing parts is essential in all OCR systems. At follow, the major activities performed in this field are mentioned.

2.3.1.2 Farsi OCR Datasets

The first effort to produce FOCR datasets was made formally in 2006 by Mozaffari, Faez, Faradji, Ziaratban and Golzan. That dataset, called Isolated Farsi Handwritten Character Data Base (IFHCDB), includes Farsi digits and separate letters. However, the samples in the IFHCDB dataset are not exactly similar to natural handwritten Farsi text, because the texts were gathered from a set of well-written high school entrance exam forms. Also, it is a non-uniform dataset with different number of samples for each class. After that, other efforts were made to produce appropriate Farsi letters, digits, words, and texts datasets such as CENPARMI (Solimanpour, sadri & Suen, 2006), Hoda (Khosravi & Kabir, 2007), Ifn/Farsi (Mozaffari, El-Abed, Margner, Faez & Amirshahi, 2008) , Islamic Azad University of Tehran/Persian Handwritten City Names (IAUT/PHCN) (Bidgoli & Sarhadi, 2008), Farsi Handwritten Text (FHT) (Ziaratban, Faez & Bagheri, 2009), and Persian Handwritten Text Dataset (PHTD) (Alaei, Nagabhushan & Pal, 2011a). Table 2.4 summarizes some of the available datasets for offline FOCR systems. In this table, the characters D, C, W, T, Q, B, and E in order are corresponding to words: **D**igits, **C**haracters, **W**ords, **T**exts, che**Q**ue legal amounts, sym**B**ols, and Dat**E**s. Also, Figure 2.5 shows some samples from the mentioned FOCR datasets.

Table 2.4 : Some datasets for offline FOCR systems

References	Scientific Name of Product	The number of writers	The used sources for gathering information	Type of Information
Khosravi et al., 2005	Hadaf	220,000	Registration Forms	D , C
Mozaffari et al., 2006	IFHCDB	---	Exam Forms	D , C
Solimanpour et al., 2006	Farsi CENPARMI	175	Special Forms	D , C , E , Q
Khosravi and Kabir, 2007	Hoda	11,942	Registration Forms	D , C
Bidgoli and Sarhadi, 2008	IAUT/PHCN	395	Postal Address	W
Mozaffari et al., 2008	Ifn/Farsi	600	Special Forms	W
Ziaratban et al., 2009	FHT	250	Forms	T
Haghighi et al., 2009	CENPARMI (Farsi)	400	Financial Documents	D , C , W , E , B
Alaei et al., 2011	PHTD	40	Historical, Scholl dictation, General texts	T

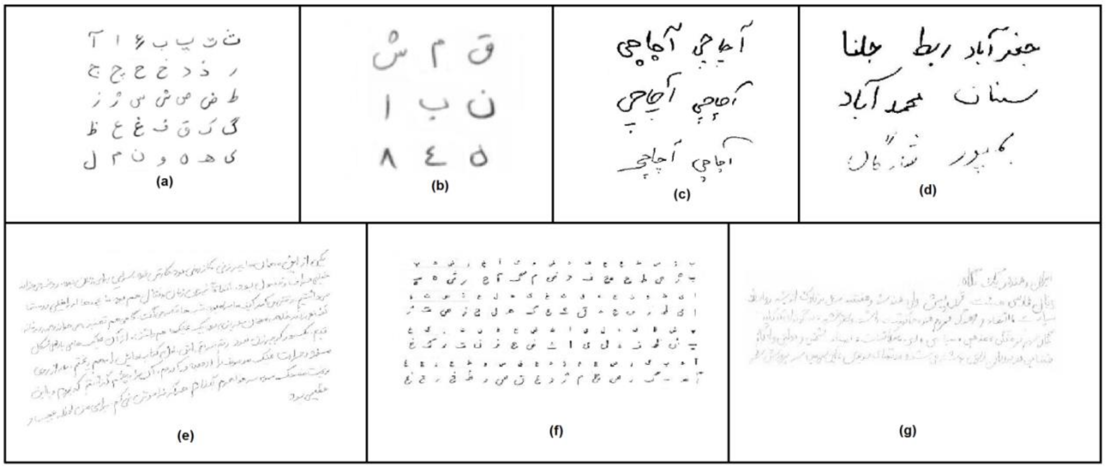


Figure 2.5 : Some samples of FOCR Datasets. (a): IFHCDB, (b): Hoda, (c): IAUT/PHCN, (d): Ifn/Farsi, (e): FHT, (f): Farsi (CENPARMI), (g): PHTD

2.3.2 Pre-processing

The performance of an OCR system depends very much upon the quality of the original data. Some defects of input documents include: low quality of the initial images, low quality of the used paper, low resolution of the used scanners and digital cameras, low quality of pens that they are used by authors for writing, and lack of accuracy in writing. Hence, it is important to pre-process the input images before they are propagated into the system to produce the system model or to perform the recognition process. The pre-processing operations will help to reduce the inherent variations of the writing methods, and create more a universal model of the input data. The main goal of these operations is to boost the quality of the input data.

In an OCR application, the pre-processing step usually includes many operations on initial raw image such as thresholding (binarization), smoothing, skew/slant correction, noise removal, thinning, baseline estimation, document structural analysis, and so on (Jumari & A.Ali, 2002). Figure 2.6 shows the outline of the main steps of a pre-processing module in an OCR system. These aspects are discussed in the following subsections, with respect to research in FOCR systems. However, it is maybe some OCR systems do not include a few parts of a pre-processing block, depending on the application and the nature of input data. For example, if all features are extracted from image skeleton, then pen width estimation is not necessary, and so on.

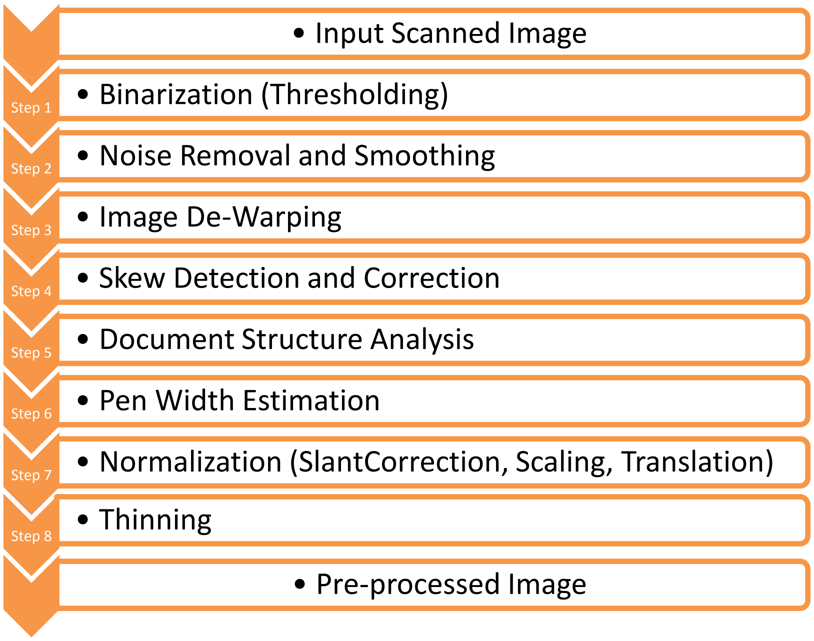

• Input Scanned Image
Step 1
• Binarization (Thresholding)
Step 2
• Noise Removal and Smoothing
Step 3
• Image De-Warping
Step 4
• Skew Detection and Correction
Step 5
• Document Structure Analysis
Step 6
• Pen Width Estimation
Step 7
• Normalization (SlantCorrection, Scaling, Translation)
Step 8
• Thinning
• Pre-processed Image

Figure 2.6: Pre-processing block components

‘Image De-Warping’ component, ‘Skew Detection and Correction’ component, and ‘Document Structure Analysis’ component did not used in this research, based on the nature of employed datasets. Hence, only the other parts are illustrated in the following sub-sections.

2.3.2.1 Binarization (Thresholding)

Generally, entered images into an offline OCR system are colored-RGB images. These images by transformation Equation 2.1 are converted to the grey level format (Gonzalez, Woods & Eddins, 2009). In this formula R, G, and B are red, green, and blue intensity of the image.

$$\text{Gray\_scale\_image\_mode} = 0.299R + 0.587G + 0.114B \tag{2.1}$$

The gray levels values are usually in range from 0 to 255. In conventional OCR systems, only binary images with two black and white values are used. Binarization process converts a gray scale image into bi-level image. In a binary image all foreground pixels have been separated from background pixels (Alirezaee, Aghaeinia, Faez & Rashidzadeh, 2005).

Extensive works have been carried out to convert a gray levels image to the binary version. The common method for this process is Otsu technique (Gonzalez, Woods & Eddins, 2009), which used the histogram of the gray values of the pixels for distinguishing foreground pixels from background pixels. The histogram of a grey level image, usually have two peaks. The larger peak is corresponding to background pixels of image and the smaller peak is corresponding to foreground pixels of an image. Minimum value between these two peaks is typically chosen as the appropriate threshold and all pixels with gray level less than threshold are classified to '0', and pixels with gray level greater than this threshold are classified to '1'. Normally, the Otsu method considers a fixed threshold for all pixels of an image. However, this method does not produce satisfactory results for complex images with multiple gray level values. Hence, Shirali, Manzuri and Shirali (2006a, 2006b) used the Otsu method technique, dynamically. They minimized the variance of total weight in some Farsi documents and found a better threshold value for the binarizing process. They achieved much better results when compared to static threshold binarization. Another version of Otsu method is as follows that an image is divided into different small squares areas, and then the different thresholds for binary image are found in each area, separately. If the dimensions of small squares areas are selected in a good manner, the results will be very good (Gupta, Jacobson & Garcia, 2007).

2.3.2.2 Noise Removals and Smoothing

One of the most important pre-processing operations is image enhancement. Usually, images that are entered into an OCR system have some undesirable parts which are generally referred to as noise in a common way. Main sources of noise generation in images are: low quality of initial images, low quality of used paper in documents, low performance of input devices, low quality of pens that the authors have written the texts by those pens, lack of accuracy in writing, and so on.

One of the most important features of Farsi letters is existing dot(s) in the letters. Eighteen of the 32 Farsi letters (more than 56%) contain one to three dot(s) above, below or in the middle of the character body (only two small letters of 26 English letters have only one dot on top of the body of the letters). Also, 23 of the Farsi letters (more than 71%) are differentiated from other letters only by their dot(s). This fact shows the importance of dots in FOCR system against the Latin OCR systems. Table 2.2 shows the different groups of Farsi letters that they have the similar bodies and therefore, they are recognized only by the number and location of their dots.

Unfortunately, the size and shape of dots in Farsi alphabet are very similar to noise pattern in images. Therefore, if a noise removal algorithm removes dots from an image, then it will result in many errors in the recognition process. Thus, the methods of noise removal for Farsi documents must be applied very carefully (Shirali & Shirali, 2008).

In order to remove noises and achieve a clean and high quality input image suitable for recognition, it is necessary that certain filters are applied on images. The following two noise removal methods have been used in this research.

a) Order Statistic Spatial Filters (OSSF)

Generally, an OSSF, as a window, covers a part of an image and then, by using a series of mathematical operations, produces a new value for the pixel located at the center of the window. One of the popular filters in this group is median filter (Gonzales, Wood & Eddins, 2009). This filter replaces the value of the pixel in center of a window by mean value of the black/white level of the neighborhood of the centered pixel by the following equation:

$$f(x,y) = \underset{(i,k) \in S_{xy}}{\text{median}} \{ g(i,k) \} \qquad (2.2)$$

In Equation 2.2, S_{xy} is a set of coordinates of points in the window size $m \times n$ concerned to filter. The point (x,y) is located in the center of that window. Also, g(i,k) is the initial value of points (x,y), and f (x,y) is the new generated value by the filter, for this point. By applying a median filter on an image, contrast of image edges has decreased, and the small holes in the image go away, and therefore, the preprocessed image is smoother.

b) Morphological Filters

Morphological operators are employed for noise removal from images. Two common practices in this way are opening and closing. Morphological opening operation is applied to open a small space between the objects within an image that are not properly in contact with each other. Morphological closing operator leads to fill the small gaps between existing objects in an image. This operator removes the small cavities and fractures of an image that might have been generated by the thinning process. Closing operator also flats a picture a little (Alirezaee, Aghaeinia, Faez & Rashidzadeh, 2005). Equations 2.3 and 2.4 respectively show the opening and closing functions:

$$A \circ B = (A \Theta B) \oplus B \qquad (2.3)$$

$$A \bullet B = (A \oplus B) \Theta B \qquad (2.4)$$

Both opening and closing functions use two operator dilation and erosion. Dilation operator has been defined mathematically as follows:

$$A \bigoplus B = \{c \;/\; c = a + b \;,\; a \in A \;,\; b \in B\} \tag{2.5}$$

Erosion operator has been defined mathematically as follows:

$$A \, \Theta \, B = \{ c \;/\; c = (B)_c \subset A \} \tag{2.6}$$

In above equations, A is the initial image and B is a structuring element. In opening operation, firstly erosion operation and secondly dilation operation are applied on images by using a specific structural element (a numerical matrix). But in closing operation, firstly dilation process and secondly erosion process are carried out. However, a drawback of using morphological opening and closing operators for image smoothing is that there is no standard approach to find the best structuring element to be implemented (Khorsheed, 2002).

2.3.2.3 Pen Width Estimation

Font width estimation (or calculation) can have a positive effect in functioning of some parts of an OCR system such as baseline detection. For this purpose, the run length values in horizontal and vertical directions are calculated and then by averaging these two values, a good approximation of the pen width will be produced (Alirezaee, Aghaeinia, Faez & Rashidzadeh, 2005) .

2.3.2.4 Normalization

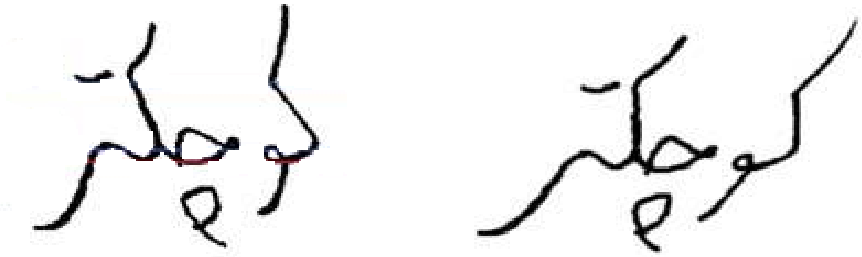
In document understanding domain, the normalization operation almost means to normalize a character to predefined dimensions, which destroys the geometrical attributes of that character. The normalization process converts an image pattern into a standardized format before it is fed into the recognition stage. It carries out some operations on the images such as de-slanting the words, rescaling images size, and moving an image to a specific coordinates. These operations are discussed in the following sub-sections, briefly.

a) Slant Correction

There are some special rules in writing printed documents, but generally handwritten texts do not follow those rules, and writers usually use their own styles. The characters shapes, forms and location of letters' dots and writing paths are examples of this subject.

One of the most obvious measurable factors of different handwriting styles is the angle between the longest strokes in a word and the vertical direction referred to as the word slant (Jumari & A.Ali, 2002). In printed documents, the letters and words are written as italic, too. However, the tilt angles for printed characters are predefined, and therefore they can easily be found and corrected. However, correcting the tilt angle in handwritten texts is difficult, because it does not have a fixed value. If an OCR system has been trained by normal samples, but it wants to recognize the slanted letters, the error recognition rate will be considerable. In this case, the extracted features in training step are different from extracted features in testing phase. This issue is far more important for Farsi texts, because some Farsi characters (or parts of them) are normally written with a slant, while another or other parts of characters are normally written vertically. Therefore, methods for finding and correcting slant in Farsi texts (especially handwritten texts) are much harder than in other languages such as English. An efficient algorithm for solving the slant problem in Farsi texts should be able to diagnose and correct the tilt of a character, while at the same time, it should be recognize the natural tilt for the rest of the body of the character and it does not change the other parts of character.

Ziaratban and Faez (2009) corrected non-uniform slant in handwritten Farsi texts in two stages. In the first stage, they employed the Prewitt filter in different directions and found some vertical-like strokes. Using the average of these vertical-like stroke tilts, they corrected the general image tilt. In the second stage, local tilts in different locations in an image are estimated and corrected, individually. They obtained acceptable results from a few handwritten samples of the IFN/ENIT dataset and also the printed version of those samples. This has been the first successful method for correcting non-uniform slant in handwritten Farsi texts, as there has been no prior report of the successful use of any other methods for this purpose. Figure 2.7 shows an example of the aforementioned technique.



a) slanted image b) Image after slant correction

Figure 2.7 : An example of slant correction technique (Ziaratban & Faez, 2009)

Hanmandlu, Murali, Chakraborty, Goyal and Roy (2003) proposed a simple method for correcting the slant of the handwritten words. First, they divided the bounding box of a character image into two equal up and down parts. Then, they calculated center of mass in those two halves. The slope of a straight line which connects these two mass point centres is considered to be the slant angle and the image is rotated in the reverse direction. This method is very fast and accurate which has been used by other researchers such as Kheyrkhah & Rahmanian (2007) to correct slant for handwritten Farsi words, too.

b) Scaling

In a printed text, the font size has fixed value until the operator changes it intentionally. However, in handwritten documents it would be very likely that a writer uses different sizes to write different letters. Therefore, different characters in different parts of a scanned image do not necessarily have the same sizes. This characteristic in Farsi languages is much more likely, because Farsi characters have very different dimensions, in nature. Therefore, it is necessary to apply various scaling algorithms that patterns with different sizes are changed to relatively equal sizes. This operation is called scaling and size normalization. In this thesis, each character is scaled to a matrix of 50×50 pixels.

c) Translation

Usually, outcome image from pre-processing stage is located in any location of m×n pixels work space. Translation process moves an image to certain coordinates in work space. One of the suitable methods for translation operation is using the concept of image's Center Of Mass (COM). COM of an image is obtained by making an average by using the all x and y pixels of that image. Equations 2.7 and 2.8 represent this concept.

$$\mathrm{C}x = \sum_{i=1}^{m} \mathrm{A}\ (\mathrm{i,j})\ /\ \mathrm{k} \tag{2.7}$$

$$\mathrm{C}y = \sum_{j=1}^{n} \mathrm{A}\ (\mathrm{i,j})\ /\ \mathrm{k} \tag{2.8}$$

In above equations , *A* is the original binary image, C_x is the center of mass of image *A* on *X* axis, C_y is the center of mass of image *A* on *Y* axis, and *k* is the total number of foreground pixels in m×n image space. After calculating the C_x and C_y of an image, this point is moved to the central point of image. As a result, all the pixels of image also move to new position in m×n space. In other word, the image is transferred to the new coordinates.

2.3.2.5 Thinning (Skeletonization)

Thinning means description of an image with fewer pixels. It is the process of minimizing the width of a line, from many pixels wide to just one pixel. Therefore, the processing time is reduced tangible and thus, the speed operation will increase. The main aim of thinning process is removing boundary pixels of a character that pixels neither are essential for preserving the connectivity of the pattern nor they represent any significant geometrical features of the pattern (Lorigo & Govindaraju, 2006). However, it should be noted that the removing pixels from an image can mean removal of some (crucial) image information. Hence, this operation may have negative impact on correct recognition rate. A good skeletonization algorithm must meet these requirements: preserve the connectivity of skeletons, coverage to skeletons of unit width, approximate the medial axis, and achieve high reduction efficiency (Nasrudin, Omar, Zakaria & Yeun, 2008). The most common thinning algorithms are based on an edge erosion technique where a window is moved over the image and a set of rules applied to the contents of the window (Gonzalez, Woods & Eddins, 2009).

In PR systems, statistical and structural features are generally extracted from a pattern to represent it. Although thinning process reduces the number of pixels in an image, but it has not effect on structural features. Thus, in this case, thinning is considered desirable operation. In contrast, removing a number of image pixels can strongly overshadow the statistical features, because the statistical features are directly related to the number of image pixels. Therefore, in this group of PR systems thinning operation will yield a wrong answer in recognition part.

Table 2.5 summarizes the mentioned pre-processing operations in some related research works in FOCR domain.

Table 2.5 : Summarization of researches in handwritten FOCR systems, based on pre-processing operations

References	**Pre-processing Activities**						
	Binarization	Noise Removal	Pen Width Estimation	Slant Correction	Scaling	Translation	Thinning
Shirali et al., (2006a, 2006b)	*						
Alirezaee et al., 2005		*	*				
Ziaratban and Faez, 2009				*			
Kheyrkhah & Rahmanian, 2007				*			
Dehghani et al., 2001	*	*			*	*	
Mowlaei et al., 2002					*	*	
Mowlaei & Faez, 2003					*	*	
Sadri et al., 2003					*	*	
Soleymani & Razzazi, 2003		*					*
Alirezaee et al., 2004a	*	*					
Pirsiyavash et al., 2005	*	*					
Vaseghi et al., 2008	*	*			*	*	
Jenabzade et al., 2011	*	*					

Discussion

- There are only a few researches with new idea, regarding to enhance the output results of pre-processing step for FOCR systems. For example, Shirali and Shirali (2008) tried to estimate the size of dots in input image, in order to save them against noise removal operation. They made an initial estimation of dot sizes, calculated the mean and variance of patterns, and found an appropriate threshold for the actual dimensions of the dots in printed Farsi texts. Finally, they removed the components that are smaller than the threshold as noise. However, it should be noted that this method is only suitable for printed texts. In the handwritten documents, the size, shape and location of the dots are not fixed, and the method cannot be applied.

- Based on the best of our knowledge, nobody try to connect the broken parts of an image together, in order to enhance the quality output of preprocessing step for Farsi digits. This aim is one of the objectives of this research.

2.3.3 Feature Extraction and Feature Selection

Feature Extraction (FE) process is one of the most important and critical stages in any PR systems such as OCR systems. The output of this stage directly influences the performance of the next stage, i.e. recognition stage (Al-Tameemi, Zheng & Khalifa, 2011).

FE involves the detection/extraction of various desired attributes (features) of an object in an image. Features are the information that it passed to the recognizer to build the system model (Parvaz & Mahmoudi, 2013), and they include the pixels densities, sample shape, or mathematical properties such as mean or mode of the input image. In FE process, a feature vector is assigned to any pattern which is fed to the system. Each feature vector discriminates a pattern from the others in features space. They also should be insensitive to irrelevant variability in the input as much as possible, and be limited in number to permit effective computation of discriminant functions and to limit the amount of training data required (Izakian, Monadjemi, Tork Ladani & Zamanifar, 2008).

In an OCR system, the objects of interest are often represented by a set of numerical features with a goal to remove the redundancy from the data. Moreover, the extracted features are expected to be invariant under affine transformation. Extracting appropriate and robust features is a basic point in an OCR system like other PR applications.

In order to construct a feature vector, some important aspects should be considered:

- A feature vector for a sample of a class should be discriminate from other feature vector for other samples in other classes as possible.
- A feature vector should be insensitive to noise, scale changes, rotation, and other probably changes related to patterns as much as possible.
- A feature vector should extract the maximum amount of pattern characteristics from input samples.
- Feature vectors should not be similar, redundant and repeatedly and they should represent different classes in feature space considerably.

It is necessary to mention that in a few cases, researchers have used the original patterns as necessary information for classification stage, instead of extracting the features from them. They have fed all the pixels of an image to recognition engine, directly (Mandal & Manna, 2011; Pradeep, Srinivasan & Himavathi, 2011). The important drawback of this approach is the large volume of input data. For an example, if a binary image has 100×100 pixels resolution, then 10,000 data value will enter as data input to the system recognition block for each sample. Whereas by applying a feature extraction technique, the volume of necessary information for recognition stage can be reduced considerably.

There are two different, but related issues, concerning the features in PR systems. The first issue pertains to find the appropriate features for a PR system, i.e. features extraction. The second issue pertains to select some of the extracted features in the first part as the final feature vector, i.e. feature reduction (feature selection, dimensionality reduction), in order to reduce dimensionality of the problem and therefore reducing time and memory usage. The following sections explain briefly these two issues.

2.3.3.1 Features Extraction in OCR Systems

Various kinds of features can be found and/or calculated in the feature extraction part in a FOCR system. Usually, features are categorized into statistical (Parvez & Mahmoud, 2013), structural (Shanbehzadeh, Pezashki & Sarrafzadeh, 2007), global transformations (Al-Khateeb, Jiang, Ren, Khelifi & Ipson, 2009), and template-based matching and correlation (Rafaa & Nordin, 2011).

a) Structural Features

The Structural Features (StrF) describe the geometrical and topological characteristics of patterns, using their global and local properties (Gonzalez, Woods & Eddins, 2009). They are the most popular features investigated by researchers in handwritten FOCR systems, because they are the intuitive aspects of writing (Khorsheed, 2002). StrF are less influenced by sources of distortions, but they are highly dependent on the style of writing (Shanbehzadeh, Pezashki & Sarrafzadeh, 2007; Khedher, Abandah & Al-Khawaldeh, 2005). Although StrF are effective, but they are not easy to extract and usually, the researchers find them heuristically. They may be extracted from each row, column, skeleton or contour of an image.

b) Statistical Features

The Statistical Features (StaF) are derived from the statistical distribution of the image’s pixels and describe the characteristics measurement of a pattern. They include numerical values computed from a part or the whole of an image. Although these features are easy to extract, they can lead the system to a wrong way, because most of them are very sensitive to noise, scale, rotation, and other changes in the patterns.

c) Global Transformation Features

Transformation process maps an image from one space to another space. These transforms usually reduce the dimensionality and order of computing in new space. Transformation processes provide feature that are invariant to global deformation like translation, dilation and rotation (Khorsheed, 2002). Some of well-known transformations are Fourier, Wavelet, Discrete Cosine transformation, and Fractal Codes. These transformations generate different features such as: Fourier descriptors, Modified Fourier Spectrum descriptors, wavelet coefficients and so on.

d) Template-based Features

Template-based features are usually created by matching pre-defined templates on graphical input data. However, they are completely data dependent, and thus, they usually cannot be transferred from one PR system to another.

Some of the most-used **structural features** (Zhang, Suen & Bui, 2004; Karic & Martinovic, 2013; Peng, Cao, Setlur, Govindaraju & Natarajan, 2013; Impedovo, 2013; Shanbehzadeh, Pezashki & Sarrafzadeh, 2007), **statistical features** (Parvez & Mahmoudi, 2013; Shah & Jethava, 2013; Chen, Beijun, Asharif & Yamashita, 2009; Abandah & Anssari, 2009; Alaei, Nagabhushan & Pal, 2010a; Noaparast & Broumandnia, 2009; Broumandnia & Shanbehzadeh, 2007; Enayatifar & Alirezanejad, 2011; Izakian, Monadjemi, Tork Ladani & Zamanifar, 2008; Rashnodi, Sajedi & Saniee, 2011; Kheyrkhah & Rahmanian, 2007; Singh, Singh & Dutta, 2010), and **transformation features** in OCR applications are shown in Table 2.6. Also, Table 2.7 summarizes feature extraction task in FOCR systems.

Table 2.6 : Some of the most used features in OCR applications

Type of Features	Some of most used features
Structural	Simple, Double and Complex Loops Loops positions, their types and their relative locations Hills and Valleys Open curves in different directions Ascenders, Descenders Number and locations of dot(s) in each character Location of dots relevant to baselines Starting, Ending, Branching, Crossing, Turning and Corner points in image skeleton Curvature and length of the image segments Length of a character segment relative to other segments Location of a character segment relative to center of mass image skeleton Image Area and Image Perimeter Aspect Ratio
Statistical	Normalized Central, Zernike, Pseudo Zernike, Fast Zernike, Legendre, Orthogonal Fourier-Mellin, Rotational, and Complex moments extracted from the whole body, only from the main body, or only from the secondary parts of an image Gradient Descriptors Pixels distribution in left, right, up and down halves of the image Image Density Mean, Mode, Variance, 2D Standard Deviation Average and Variance of X and Y changes in portions of the image skeleton Ratio of horizontal variance histogram to vertical variance histogram Ratio of up-halve variance to down-halve variance of an image Thinness Ratio The ratio of pixel distribution between two or more parts of the image Center of Mass (COM) (Center of Gravity) Centroid distance Radial coding Top, Bottom, Left and Right Profile histograms Number of Horizontal (Row) and Vertical (Column) transitions Number of Modified Horizontal and Vertical transitions Outer and Inner Contour directional Chain Code histograms Normalized, and Modified Contour Chain Code Fractal, Shadow Code Descriptors Energy of original image Number of specific points such as end, branch, and cross points Number of connected components Relative location of start and end points of an image skeleton Pen width and Line height Baselines positions Histogram of slopes along contour Skeleton-based N-degree directional descriptors
Transformation	M-band packet wavelet coefficients Fourier, DCT, and Radon coefficients

Table 2.7 : Summarization of researches in handwritten FOCR systems, based on feature extraction operation

Researchers	Features	No. of Features	Data Type		
			Digit	Character	Word
Shirali et al., 1994	Zernike moments	45	*		
Shirali et al., 1995	Shadow code descriptors	32	*		
Hosseini and Bouzerdoum, 1996	Number of crossing between digit body and horizontal and vertical raster lines	10	*		
Mowlaei et. al, 2002	Harr wavelet coefficients	64	*	*	
Mowlaei and Faez, 2003	Harr wavelet coefficients	64	*	*	
Sadri et al., 2003	Derivative of 4 different views from 4 main directions using counting the number of background pixels between border and outer boundary.	64	*		
Soltanzadeh and Rahmati, 2004	Outer profiles of images at multiple orientation, Crossing counts, Projection histograms	257	*		
Mozaffari et al., 2004a	Fractal codes	64	*	*	
Mozaffari et al., 2004b	Fractal codes, Harr Wavelet transform	64	*	*	
Mozaffari et al., 2005a	Fractal code	240	*		
Mozaffari et al., 2005b	Average and variance of X and Y changes in different portion of the skeleton, End points, Intersection points	75	*		
Mozaffari et al., 2005c	Fractal code	240	*		
Mozaffari et al., 2005d	Fractal code	64	*	*	
Harifi and Aghagolzadeh, 2005	Pixels density in 12-segment digit pattern, Moment inertia, Center of mass	16	*		
Ziaratban et al., 2007a	Position of the best occurred matching in the horizontal and vertical coordinate template, Amount of the best matching template. Templates : slanted lines, T junction, up, down, right and left curvature and so on.	60	*		
Alaei et al., 2009a	Chain code direction frequencies of image contour	196	*		
Alaei et al., 2009b	Modified chain code direction frequencies in contour. Modified horizontal and vertical transition levels	198	*		
Salehpour and Behrad, 2010	Automatic feature extraction using PCA	20, 30, 40, 50	*		
Enayatifar & Alirezanejad, 2011	Pixels accumulation, Pixels Direction	48	*		
Alirezaee et al., 2004a	Relative energy of eroded versions respect to original image, Displacement of center of mass. Minimum and maximum eigenvalue.	63		*	
Alirezaee et al., 2004b	Invariant central moments	7		*	
Shanbehzadeh et al.,	Number of component in each character,	78		*	

2007	Number and location of dots relevant to baseline, Number of pixels in each frame cell, Center of mass of each cell				
Ziaratban et al., 2008b	Terminal points, Two-way branch points, Three-way branches points	32, 40, 64, 108		*	
Gharoie and Farajpoor, 2009	All pixels of an image	900		*	
Alaei et al., 2010	Modified chain code direction frequencies of the contour	196		*	
Jenabzade et al., 2011	Wavelet coefficients from outer border, Central moments	134		*	
Rajabi et al., 2012	Zoning densities, crossing count, outer profiles	315		*	
Alaei et al., 2012	Dimensional gradient	400		*	
Dehghan et al. 2001a	Slope, curvature, number of active pixels and slope and curvature of each section extracted from the contours.	20 × number image's frames			*
Dehghani et al. 2001	Pixels densities in various regions, contour pixels, angle of line passing through the first and end point in each image parts, ….	------			*
Safabakhsh and Adibi 2005	Moments, Fourier descriptors, Number of loops, Aspect ratio, Pixel densities, Position of right and left connections End, Junction, Branch, and Crossing Points	9			*
Broumandnia et al. 2008	Wavelet packet transform coefficients	16, 32, 96, 128, 160			*
Vaseghi et al. 2008	Statistical Density Values	4 × number of image frames			*
Mozaffari et al. 2008b	Black – white pixel transition	10 × number of image windows			*
Bagheri and Broumandnia 2009	Zernike moments	9, 25, 49, 72, 100, 182			*
Bahmani et al. 2010	Wavelet coefficients extracted from smoothed word image profile in four up, down, left and right directions.	200, 400			*

2.3.3.2 Features Selection (Features Reduction, Dimensionality Reduction) in OCR Systems

Various kinds of features are computed and/or extracted in the feature extraction step of an OCR system. Some of the extracted features, however, might correspond to very small details of the patterns, or might be a combination of other features (non-orthogonal features), while some others might not have any efficacy in the recognition stage (Shayegan & Chan, 2012). Irrelevant or redundant features may degrade the recognition results, reduce the speed of learning of the algorithms, and significantly increases the time complexity of the recognition process (Azmi, Pishgoo, Norozi, Koohzadi & Baesi, 2010). Hence, using all extracted features does not always produce the desired results, and could also increase the time complexity of the recognition process (Shayegan & Aghabozorgi, 2014a). Therefore, following the feature extraction process, the issue of Feature Selection (FS) (Feature Reduction, Dimensionality Reduction) arises.

FS is typically a search problem to find an optimal subset with *m* features out of the original *M* features. In other words, FS is a process for excluding irrelevant and redundant features from the feature vector. The final goal in FS operation is reducing system complexity; reducing the overall processing time, and increasing system accuracy (Guyon & Elisseeff, 2003).

In this respect, some features subsets selection algorithms have been proposed. According to the criterion function used for finding one *m* members subset out of 2^M possible subsets (*M* is the number of initial features), two general categories have been introduced for this important task: Wrapper algorithm and Filter algorithm (Dash & Liu, 1997). In the Wrapper algorithm, the classifier performance is used to evaluate the performance of a features subset. In the Filtering algorithm, the features evaluation function is used rather than optimizing the classifiers performance. In this category, the best individual features are found one by one. However, the *m* best features are not the best *m* features (Peng, Long & Ding, 2005). Usually, the Wrapper methods are slower, but perform better than the Filter methods.

Dimension reduction (feature selection) is one of the objectives of this research. Hence, the various methods and researches are reviewed in more details. Based on the removing strategies, the feature selection methods are categorized into three general groups, as follows:

a) Sequential Backward Selection

One of the methods for decreasing the number of features is Sequential Backward Selection (SBS) technique (El-Glaly & Quek, 2011). In this approach, features are deleted one by one, and the system performance is measured to determine the feature performance.

b) Genetic Algorithms

Another method for selecting a limited number of features out of a large set of features comprises the random search methods such as Genetic Algorithms (GA). GA keeps a set of the best answers in a population (Bahmani, Alamdar, Azmi & Haratizadeh, 2010).

Azmi, Pishgoo, Norozi, Koohzadi and Baesi (2010) used GA in a handwritten FOCR system. Their proposed system initially had 81 features with an accuracy of 77%. After applying GA, the number of features was reduced to 55 and the accuracy increased from 77% to 80%.

Kheyrkhah and Rahmanian (2007) employed GA to optimize the number of initial extracted features in a handwritten Farsi digit recognition system. They showed that not only all extracted features are not useful in classification, but they also reduce the recognition accuracy and increase the system learning time. They first selected a random subset of the initial features set, trained a Bayesian classifier with this features subset, and reported recognition errors. In the next stage, 50% of the previous features and 50% new features made the a features subset, and recognition rate was computed and compared with the old results. Comparing these two recognition rates led to selection of some features. This operation was repeated until appropriate results were achieved. Their system reduced the number of features from 48 to 30 and it increased recognition rate from 75% to 94%, but the elapsed time for this significant increase was significant.

c) Principal Component Analysis

The third method for feature selection operation is a group of statistical methods, such as Principal Component Analysis (PCA) (Song, Yang, Siadat & Pechenizkiy, 2013), and Random Projection (RP) (Van der Maaten, Postma, & Van Den Herik, 2009), that have been applied to find important patterns in high-dimension input data. PCA is a pre-processing step in recognition applications which involves converting a correlated features space to a new non-correlated features space. In the new space, features are reordered in decreasing variance value such that the first transformed feature accounts for the most variability in the data. PCA is based on the statistical representation of a random variable, and has been widely used in data analysis (Bouchareb, Hamidi & Bedda, 2008). A brief description of PCA technique has been explained in Appendix III. The discussion about these feature selection techniques is postponed to the end of this chapter.

2.3.4 Classification (Recognition)

The classification process includes methods for assigning a new instance, using its features vector, to one of the existing classes in the pattern space. In other word, by using one (or more) classifier(s), unlabeled testing samples are classified to a labeled training set. Generally, classification is carried out by minimizing the distance between new instance features vector and generated features vectors in the training step.

The available classification methods are usually categorized into following general groups (Lorigo & Govindaraju, 2006): Template Matching; Statistical Approach; Decision Trees; Neural Networks (NNs); Hidden Markov Models (HMMs); and Support Vector Machines (SVMs). Nevertheless, it has been shown that some techniques of one group used in other groups. Also, if a single classifier fails to yield high performance, several classifiers (Combined Methods) may be combined to give acceptable results (Alginahi & Siddiqi, 2010). In this thesis, *k*-NN, Neural Network, and Support Vector Machines classifiers have been used in recognition block in different experiments. Hence, the following sections demonstrate the mentioned used classifiers, briefly.

2.3.4.1 *k*-NN

The k Nearest Neighbor (*k*-NN) is one of the well-known classifiers in statistical classifiers group. The *k*-NN method is a fast supervised machine-learning algorithm which is used to classify the unlabeled testing set with a labeled training set (Alavipour & Broumandnia, 2014). In order to classify a new object, 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 image is then found based on the minimum difference between the testing object image and the training samples.

The *k*-NN algorithm can be described by using the following equation (El-Glaly & Quek, 2011):

$$y(d_i) = \arg\max \sum_{xj \in k-NN} Sim(d_i, x_j)\, y(x_j, c_k) \quad (2.9)$$

d_i : testing sample

x_j : one of the neighbors in the training set

$y(x_j, c_k) \in \{0, 1\}$ indicate whether x_j belong to class c_k
$Sim(d_i, x_j)$: Similarity function for d_i

Finally, the class with maximal sum of similarity will be selected as testing sample class.

2.3.4.2 Neural Networks

An Artificial Neural Network (ANN) is a non-linear system with a large number of internal connections which it be characterized based on a particular network topology. Learning methodology and characteristic of the neurons define the topology of a NN. It usually includes a large number of neurons and a high degree mesh of connectivity between neurons. In a NN, each neuron is considered as a simple processing element with a very simple task.

There are usually three general layers in a NN: an input (initial) layer, one (or more) hidden (intermediate) layer(s), and a final output layer as shown in Figure 2.8. Neurons are the smallest units in each NN which connect the three layers of NN together. They are connected together by one or more links. Each link between two neurons has an especial weight (value). These weights are found in training phase of the NN, using training data. The number of nodes in input, hidden and output layers will determine the network structure. The number of neurons in input layer is usually equal to number of features in features vector. The number of neurons in output layer is usually equal to number of available classes in patterns space. However, the number of hidden layers and number of neurons in each hidden layer are determined experimentally. The best network structure is normally problem dependent and hence structure analysis has to be carried out to identify to optimum structure (Gharoie Ahangar & Farajpoor Ahangar, 2009).

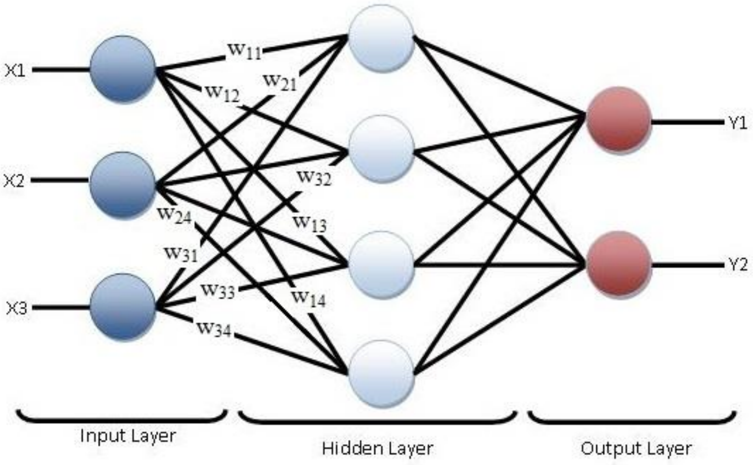


Figure 2.8 : Sample form of a neural network

NNs are one of the most successful recognition engines in the PR domain, especially in OCR research (Patel, Patel & Patel, 2011), because of their simplicity, generality, and good learning ability (Singh, sing & Dutta, 2010). As a result, many researchers have used different variations of NNs for online and offline OCR applications. The most commonly used family of NNs for FOCR is feed-forward network, which include Multi-Layer Perceptron (MLP) (Mowlaei, Faez & Haghighat, 2002; Alirezaee, Aghaeinia, Ahmadi & Faez, 2004a; Alirezaee, Aghaeinia, Faez & Rashidzadeh, 2005; Mozaffari, Faez & Rashidy Kanan, 2004a; Noaparast & Broumandnia, 2009; Enayatifar & Alirezanejad, 2011; Gharoie Ahangar & Farajpoor Ahangar, 2009; Kamranian, Monadjemi & Nematbakhsh, 2013; Ziaratban, Faez & Faradji, 2007), and Radial Basis Function (RBF) (Bahmani, Alamdar, Azmi & Haratizadeh, 2010) networks.

2.3.4.3 Support Vector Machines

Recently, there has been considerable interest in the Support Vector Machines (SVMs) classification technique. The basic idea of SVM, utilized in PR domain, is to construct a hyper-plane, as decision plane, which separates the positive and negative patterns with the largest margin. Originally, SVMs were developed for two-class problems. They looked for the optimal hyper plane which maximized the distance (margin) between nearest examples of both classes, named support vectors. In other words, a SVM selects a small number of critical boundary samples from each class and builds a linear discriminant function for them (Mozaffari, Faez & Rashidy Kanan, 2005).

The decision function derived by a SVM classifier for a two-class problem can be formulated using a kernel function K (x, x_i) of a new sample $\boldsymbol{x}$ and a training sample $\boldsymbol{x_i}$, as follows (Izabatene, Benhabib & Ghardaoui, 2010):

$$f(x) = \sum_{i \in SV} \alpha_i \, y_j \, K(x_i , x) + \alpha_0 \qquad (2.9)$$

where SV is the support vector set (a subset of training set) and $y_i = \pm 1$ the label of sample x_i . The parameters $\alpha_i \geq 0$ are optimized during the training process. A SVM uses a kernel K to construct linear classification boundaries in higher dimensional spaces. The linear SVM can be extended to a non-linear classifier using different kernel functions like polynomial and Gaussian. Table 2.8 shows some common kernels which are used in SVMs classifiers.

Table 2.8 : Different kernels in SVMs

Kernel Name	**Kernel Function**
Linear	$K(X_i , X_j) = X_i^T X_j$
Polynomial	$K(X_i , X_j) = (X_i^T X_j + 1)^d$
Radial Basis Function (RBF)	$K(X_i , X_j) = \exp [-\|\|X_i^T - X_j\|\|^2 / 2\sigma^2]$
Exponential RBF	$K(X_i , X_j) = \exp [-\|\|X_i^T - X_j\|\| / 2\sigma^2]$
Perceptron	$K(X_i , X_j) = \tanh (\lambda X_i^T X_j + \Theta)$

Highly generalization and good performance of two-class SVMs have encouraged researchers to extend them for solving multi-class problems. One of common approach for this goal is using one-against-all (one-against-other) technique. In this approach, if there exist *n* different classes, in the firsts step, one class is put in the first group and all n-1 classes are put in the second group and the system is trained to recognize the class in group '1'. Then, this process is repeated, but this time one of the classes in group '2' is considered as a new group against n-2 other classes. This process is repeated till all classes separate from each other.

Recent results in PR applications, particularly in OCR application, have shown that SVMs have become one of the most powerful classifiers, and have achieved superior recognition rates when compared with other classifiers (Alaei, Pal & Nagabhushan, 2009; Mowlaei & Faez, 2003; Sadri, Suen & Bui, 2003; Salehpor & Behrad, 2010; Rajabi, Nematbakhsh & Monadjemi, 2012). Table 2.9 shows related researches of utilizing a recognition engine in FOCR systems.

Table 2.9 : Summarization of researches in handwritten FOCR systems, based on classification engine

References	Classification Engine				
	k-NN	Neural Networks	SVM	HMM	Decision Tree
Alavipour & Broumandnia, 2014	*				
Rajabi et al., 2012	*	*	*		
Gharoie & Farajpoor, 2009		*			
Mowlaei et al., 2002		*			
Alirezaee et al., 2004a, 2005		*			
Mozaffari et al., 2004a		*			
Mozaffari et al., 2004b			*		
Mozaffari et al., 2005c	*	*	*		
Soltanzadeh & rahmati, 2004			*		
Noaparast & Broumandnia, 2009		*			
Enayatifar & Alirezanejad, 2011		*			
Kamranian et al., 2013		*			
Ziaratban et al., 2007		*			
Bahmani et al., 2010		*			
Mozaffari et al., 2005			*		
Alaei et al., 2009, 2010a			*		
Mowlaei & Faez, 2003			*		
Sadri et al., 2003		*	*		
Salehpor & Behrad, 2010			*		
Dehghani et al., 2001				*	
Dehghan et al., 2001b				*	
Alirezaee et al., 2004b	*				
Pirsiyavash et al., 2005		*			
Safabakhsh & Adibi, 2005				*	
Vaseghi et al., 2008				*	
Ebrahimpor et al., 2010		*			
Jenabzade et al., 2011		*			*
Rashnodi et al., 2011			*		
Mousavinasab & Bahadori, 2012	*				

Discussion

In order to compare different classifiers together, some important issues should be considered:

- Although the *k*-NN classifier is a fast supervised classification method, but the number of training samples and the number of features per samples have the direct effect on recognition time. In other word, high CPU cost when a large number of samples are used in the training process is the main drawback of this classifier. Hence, this powerful recognition engine faces to problem in handling very large datasets with a huge number of attributes.
- Although, design and implementation of NNs is almost a simple task, and currently, there are some efficient tools for doing that, but it should be mentioned that if a new class is added to a predesigned NN, it should be necessary to train the network again and recalculate the new weights and the number of neurons in the hidden layer which this is a time-consuming process. Also, there is no specific way of finding the correct model of NN (Patel, Patel & Patel, 2011).
- It is appropriate to utilize HMMs as the recognition engine in implementing a holistic OCR system with a small number of classes. The accuracy of this classifier group, however, will dramatically decrease with an increase in the number of classes. Handwritten documents are clustered in very large PR systems, because of the wide variety of writing styles. Hence, employing HMMs in OCR applications is restricted only to small vocabulary sets and special applications such as bank cheque processing, or mailing operations.
- The findings from many researches indicate that SVM is superior to other classification methods (Section 2.3.4.3). The linear SVM can be extended to a non-

linear version, using different kernel functions such as Polynomial, Gaussian, Radial Basis Function (RBF), and Perceptron. Nonetheless, when SVM is used in the classification stage of a PR system, the required computational time for classifying input data grows with the square of the number of samples in the training dataset (Zhang, Suen & Bui, 2004).

2.3.5 Post-processing

Commercially available OCR products use additional information, tools, and algorithms to reduce errors at the classification stage, for example, using a dictionary (or a lexicon) to spell-check a recognized word, or applying grammatical rules to correct some misrecognized characters (words).

Mehran, Shali and Razzazi (2004) proposed a statistical modeling method to correct Farsi names in the post-processing block of a FOCR system. They used a statistical grammar to add a new name to a dynamic Farsi dictionary. Using this approach, they improved the accuracy from 89.04% to 92.23% for initial Persian names, and from 77.78% to 90.85% for a few limited Persian surnames.

Ziaratban, Faez and Ezoji (2007) tried to recognize legal amount (textual format) on a bank cheque to confirm or correct recognized courtesy amount (numerical format) available on the same cheque, because most of industrial cheque amount recognition systems, in the market, rely only on the recognition of the courtesy amounts. They succeeded in improving the recognition rate from 85.33% (before using legal amount) to 99.31% (after using legal amount) as a post-processing operation.

2.4 Related Works in Handwritten FOCR Domain

There are a large number of valuable researches with acceptable results to recognize **printed Farsi** texts (Broumandnia & Shanbehzadeh, 2007; Izakian, Monadjemi, Tork Ladani & Zamanifar, 2008; Khosravi & Kabir, 2009; Pirsiavash, Mehran & Razzazi, 2005; Pourasad, Hassibi & Banaeyan, 2011; Salmani Jelodar, Fadaeieslam & Mozayani, 2005; Zand, Naghsh Nilchi & Monadjemi, 2008), to recognize **printed Arabic** documents (Al-A'ali & Ahmad , 2007; Al-Tameemi, Zheng & Khalifa, 2011; Mahmoud & Mahmoud, 2006; Khorsheed, 2007), and also to recognize **handwritten Arabic** charcaters (Abandah, Younis & Khedher, 2008; Abandah & Anssari, 2009; Abuhaiba, 2006; Al-Hajj, Likforman & Mokbel, 2009; Al-Khateeb, Jiang, Ren, Khelifi & Ipson, 2009; Al-Khateeb, 2012; Bouchareb, Hamdi & Bedda, 2008; El-Abed & Margner, 2007; Elglaly & Quek, 2011; Khedher & Abandah , 2002; Khedher, Abandah & Al-Khawaldeh, 2005; Sabri & Sunday, 2010; Dinges, Al-Hamadi, Elzobi, Al-Aghbari & Mustafa, 2011; Khalifa, Bingru & Mohammed, 2011; Mahmoud & Olatunji, 2010). In this thesis, most of the mentioned researches were studied and some techniques and algorithms were tested. However, this thesis has been focused on **handwritten Farsi letters and digits** recognition. Hence, in this section, only the researches which have been carried out in handwritten Farsi letter recognition domain are reviewed.

Dehghani, Shabani and Nava (2001) used contour of projection for designing a FOCR system. They applied some common pre-processing techniques such as median and morphological filtering, binarization, scaling and translation on character images. They projected the image in horizontal and vertical direction and then obtained the chain code of projections contour. Slope, curvature, and number of active pixels in different parts of image contour were extracted as features. They employed two HMMs for modeling horizontal and vertical projection of each character and achieved to 92.76% and 71.82% recognition rate on the training and testing samples respectively.

Dehghan, Faez, Ahmadi and Shidhar (2001b) introduced a holistic FOCR system which utilized a discrete HMM classifier. The employed features in their system were histogram of slopes along contour of the character images. The patterns were the name of 198 cities in Iran which are used in mailing system. The proposed system achieved to word-level recognition rate of up to 65% without using contextual information from the datasets.

Mowlaei, Faez and Haghighat (2002; 2003) computed Harr wavelet coefficients (discrete version of wavelet transform) as a features set for recognition isolated handwritten Farsi letters and digits. First, they found the bounding box of each character, and then normalized the image dimensions to 64×64 pixels in order to scale normalization. Also, to achieve invariance respect to translation, scale and stroke width, normalization algorithms preceded the feature extraction stage for each of these parameters. By removing secondary parts of letters, such as dots, they categorized the letters into 8 classes. Pyramid algorithm was applied on each pre-processed image to reduce the size of input images. They finally made a 64-dimensional features vector for each image. They also employed a feed forward NN using back propagation learning rule as classifier. They used the proposed system to recognize 579 cities names, and also postal code in Iran. It is necessary to mention that only eight digits out of 10 possible digits are used in zip codes in Iran. Training and testing dataset were gathered from 200 people; include 3840 digits and 6080 letters. They achieved to 92.33% and 91.81% accuracy for testing digits and letters, respectively.

In 2003, Sadri, Suen and Bui proposed an OCR system for recognizing handwritten Farsi and Arabic digits. In the first stage, they applied normalization operations as pre-processing on the images, and finally, each image was changed to a 64×64 pixels image invariant to size and translation. In features extraction block, they counted the number of background pixels between border and outer boundary of any image, from four different views of any image, and created a histogram. They considered each of these histograms as a curve, and then calculated derivative of them. To reduce the volume of features, they selected 6 samples from each derivative curve. Finally, a 64-dimensional features vector was created for each image. Using SVMs with RBF kernel, they achieved 94.14% recognition accuracy. For the sake of classifiers comparison, they employed a MLP-NN classifier with two hidden layer, too. But the outcome results in this part were weaker than SVMs related results; 91.25%. They used digits part of CENPARMI dataset as a benchmark standard dataset with 7390 and 3035 samples for training and testing, respectively.

Soleymani and Razzazi (2003) presented a FOCR system to recognize isolated handwritten letters. Their system found letter boundaries, removed noises, deleted the secondary parts of letters, and extracted the skeleton of each letter. They achieved 96.4% recognition rate on a dataset of 220,000 handwritten forms, which they were created by more than 50,000 writers. However, it is mentioned that the forms were written in a good manner and with a high accuracy duo to their natures.

Alirezaee, Aghaeinia, Ahmadi and Faez (2004a) searched for finding an appropriate features set to recognize handwritten middle age Persian (viz. Pahlavi) characters. This alphabet has only 16 isolated characters. After some pre-processing operations such as noise removal and thresholding, they applied morphological erosion operator with many structure elements, variable lengths and directions on the images, because it is evident that different structure elements have different effects on the character images. They made a 63 element features set, include some relative energy of eroded versions respect to original image, displacement of center of mass, minimum and maximum eigenvalue and so on. They employed a feed forward NN with one hidden layer and 150 neurons in hidden layer as classifiers. They finally achieved to 97.61% accuracy in their research. In another effort, they selected a set of invariant moments as features and minimum mean distance and also *k*-NN as classifiers (Alirezaee, Aghaeinia, Ahmadi & Faez, 2004b). The best result which they achieved was 90.5% correct classification rate.

Mozaffari, Faez and Rashidy Kanan (2004a; 2004b) proposed a new method for recognition isolated handwritten Farsi letters and numerals to recognize the mail code and cities names for Iran post ministry. In feature extraction step, they extracted a 64-dimension of fractal codes as features vector. Similar to (Mowlaei & Faez, 2003), they categorized Farsi isolated letters into eight groups. Since fractal codes are so sensitive to affine operation, therefore they applied some pre-processing operation for location invariability and scale normalization. But their method is still sensitive to rotation. In classification part, they employed two MLP-NNs, the first one for digits recognition and the second one for letter recognition. By the nature of fractal codes, their method was robust to image scale and size changes. For train and test the system, they used the same dataset in (Mowlaei & Faez, 2003). They obtained 91.37% and 87.26% accuracy for digits and letters, respectively.

One of the best results in handwritten Farsi digit recognition was achieved by Soltanzadeh and Rahmati (2004). Unlike other researchers that used image profiles for extracting features, they used the outer profile of digits images at multiple orientations such as top, down, left, right, diagonal, and off-diagonal as main features. The profiles count the number (distance) of pixels between the boundary box of a character image and the edge of character. The profiles describe the external shapes of characters to facilitate differentiation among a large numbers of objects. Figure 2.9 illustrates a sample Farsi digit '۴' ('4') and its four main profiles.

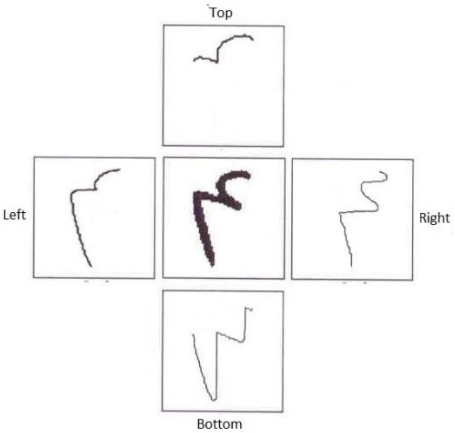


Figure 2.9 : Farsi digit '۴' ('4') and its main profiles

Although profiles are dependent on the image dimensions, they become scale-independent by normalizing the images. After normalizing the profiles, the researchers used the normalized profiles directly as features. Using only outer profiles causes the inner shape information of characters are lost, therefore, the researchers also used 'normalized crossing counts' and 'projection histograms of the image' as complementary features. The total number of features is 32×n+1, where n is the number of orientations for calculating the outer profiles. A dataset were created by 90 persons including 4974 train samples and 3939 test samples. They employed a SVM classifier one time with polynomial kernel and another one with RBF kernel in one-rest method. The best result they obtained was 99.57% accuracy using eight orientation profiles (i.e. 257 features) and using RBF kernel.

Mozaffari, Faez and Ziaratban (2005b; 2005c) used fractal code as features vector and *k*-NN classifier for handwritten zip code recognition. Similar to previously their works, they applied the same pre-processing operations on the same previous dataset, and they achieved to 86.3% accuracy. In another part of that research, they introduced fractal transformation classifier for OCR applications. They normalized and reduced the number of fractal features by using PCA techniques to 240. Method of classification was based comparing fractal code representation of a new sample with fractal code representation of all training samples. They obtained 90.6% recognition rate in final stage. They evaluated the performance of using fractal codes as features, by using RBF-NN and also SVM as classifiers. They showed SVM have better recognition rate and better generalization ability than RBF-NN classifier, but it takes more time to be trained.

Pirsiyavash, Mehran and Razzazi (2005) employed a set of NNs to recognize isolated handwritten Farsi letters. In the first step, they applied pre-processing operations noise removal, binarization and skew detection on input texts. They then categorized all letters to 13 separate groups. Central moments, ratio of horizontal variance to vertical variance, ratio of black pixels in up halve to button halve for each letter image, and so on were extracted from input images as features. In the first recognition stage, a MLP-NN classified a letter to one of these 13 classes. In the second recognition stage, they trained four other NNs which each of them classify members of each group. All the networks had three layers with 14 neurons in hidden layer. The final accuracies of their system were 77.2% and 84.4% without and with using a dictionary for post-processing operation.

Safabakhsh and Adibi (2005) employed a HMM as the recognition engine in order to recognize handwritten Farsi words in special writing style, Nasta'aligh. They removed ascenders and descenders to avoid some recognition errors. However, there are a lot of vertical overlaps and also slanted letters sequences in Nasta'aligh writing style. Hence, finding the baselines and order of characters is a difficult task in this style. The proposed system over-segmented words into pseudo-characters using local minima of upper contour. Fourier descriptors, number of loops, aspect ratio, pixel densities, and position of right and left connections were used in this research as features. They used a lexicon of 50 words, including all isolated letters and compound forms of letters. Seven writers produced the training and testing datasets. The recognition rates for two writers which wrote the words for testing from lexicon were 69% and 91% with 5 and 20 iterations of recognition steps, correspondingly. But, the recognitions were 52.38% and 90.48% on 21 words out of lexicon with 5 and 20 iteration of the recognition step, respectively.

Shanbehzadeh, Pezashki and Sarrafzadeh (2007) tried to recognize isolated handwritten Farsi letters by combining two groups of features, include three and 75 features. Some of those features were structural like: number of component in each letter, number and location of dots relevant to baseline and so on, and some features were statistical information such as: number of pixels in each frame cell, center of mass of each cell and so on. They used a dataset with 3000 letters, 60% of samples for training and 40% for testing the system. They applied vector quantization technique in recognition phase. Using all 78 features, they achieved to 87% accuracy.

Ziaratban, Faez and Faradji (2007) extracted some language-based features for handwritten Farsi digits recognition. For any image, they found three features; i.e. the position of the best occurred matching in the horizontal and vertical coordinate and also the amount of the best matching. Therefore, final features vector has a length three times more than the number of templates. They chose 20 templates like slanted lines, T junction, up, down, right and left curvature and so on, heuristically. They tested their system on a dataset with

6000 sample for training and 4000 samples for testing. They also employed a NN-MLP as a classifier and succeeded to achieve 97.65% accuracy.

Another effort for recognizing handwritten Farsi cities names, in postal address, were carried out by Vaseghi, Alirezaee, Ahmadi and Amirfattahi (2008). After applying pre-processing steps including binarization, noise removal and scale normalization, they extracted a 4-dimensional features vector from a set of overlapped vertical fixed-width frames of any image using the sliding window technique. Their dataset includes 6000 image of 198 cities names. Overall, they used 400,000 frames to generate a codebook for each class. They used vector quantization and HMM for recognition and could achieved to 95% recognition rate at the best conditions.

In other effort, Alaei, Nagabhushan and Pal (2009a; 2009b) computed two type of features set, modified chain code direction frequencies from contour of each handwritten Farsi digit image (196 features) and modified horizontal and vertical transition features (2 features), for recognition the handwritten Farsi digits. They did not use any pre-processing techniques. Therefore the speed of recognition is more than similar systems which use pre-processing operations. In recognition stage, they employed a SVM with Gaussian kernel. Finally, they attained 99.02% accuracy using Hoda dataset for training and testing the system.

Noaparast and Broumandnia (2009) used Zernike moments as features to overcome on scale and rotation difficulties, for recognizing 28 handwritten Farsi cities names. At first, using different pre-processing techniques, enhancement was carried out on an input image. Without any segmentation, they used a holistic approach for recognition. For classification stage, they employed a MLP-NN with one hidden layer. The numbers of input features to network were 9, 25, 49, 72, 100 and 182 for each of cities names to investigate the effect of the number of the features on accuracy. Each image with eight different angles of rotations was processed. The number of neurons in hidden layer was calculated 50 by trial and error method. The maximum efficiency of this method had been 98.8%.

Gharoie Ahangar and Farajpoor Ahangar (2009) employed a MLP-NN with 24 neurons in hidden layer for recognition of handwritten Farsi characters. First, they applied smoothing, thresholding and skeletonization operations on the input images. No feature extraction was carried out on the images, and the pixels of an image directly fed into the input layer. The accuracy for this system was 80% for just 125 test sample.

To recognize the isolated handwritten Farsi letters, Alaei, Nagabhushan and Pal (2010a) proposed a two-stage SVM based classifier. They categorized similar shape letters into eight groups to overcome the problem of confusion between main body similar letters. For this clustering, they made a 49-dimensional features vector. As a feature extraction technique, they used modified chain code direction frequencies of the contour pixels and compute a 196-dimensional features vector. For discriminating the pattern in the eight first groups with more than one class, they employed another SVM in the second stage. In both stages, they used one-against-other SVMs approach. By testing this system on IFHCDB dataset with 36,682 samples for train and 15,338 samples for test, they achieved to 96.68% correct recognition rate. They also showed that SVM with Gaussian kernel can produce better results, when compared to the linear and polynomial kernels.

Bahmani, Alamdar, Azmi and Haratizadeh (2010) designed a holistic Farsi OCR system for recognizing 30 handwritten Farsi names. Common pre-processing operation such as binarization and scaling were carried out on the words images. The features in their system were wavelet coefficients extracted from smoothed word image profile in four directions up, down, left and right. They employed a RBF-NN as a classifier. Using 1D discrete wavelet transform, they reduced the number of features from 400 to 200 in features vector. The best reported result for their system was 87.6% with 96 neurons in hidden layer and by using Euclidian distance in competitive layer unit.

Jenabzade, Azmi, Pishgoo and Shirazi (2011) employed an MLP-NN classifier with one hidden layer to recognize handwritten isolated Farsi letters. In beginning, they applied some pre-processing operation for binarization, smoothing and noise removal. For feature extraction stage, they extracted the wavelet coefficients from the outer border of letter images and re-sampled them in order to normalize the number of features. They finally created a 134-dimensional features vector for every letter image. In this experiment, they succeed to achieve 86.3% accuracy for testing samples. To obtain better results, they divided the input letters to five categories based on the number of components of each letter. Then, they calculated central moments for each category as features. By using a decision tree classifier, they succeeded in improving the accuracy to 90.64%. They used their own dataset including 6,600 samples (200 samples for each letter).

Rashnodi, Sajedi and Saniee (2011) used discrete Fourier transform coefficients as features set and a SVM engine with Gaussian kernel as classifier to recognize handwritten Farsi digits. After pre-processing operations, they made a 154-dimensional features vector for each digit. The features are the first 25 Fourier coefficients of image contour, average angles distance pixels, aspect ratio, and so on. Finally, they achieved 99% accuracy.

Tables 2.10, 2.11, and 2.12 summarize in chronological order, some of the researches that have been conducted for handwritten Farsi digits, letters, and words recognition. These tables include the type and the number of extracted features, and the system accuracy, too.

Table 2.10 : Some Farsi handwritten numerals recognition researches

Researchers	**Features**	**No. of Features**	**Recognition Engine**	**Accuracy (Best Case)**
Shirali et al., 1994	Zernike moments	45	NN	--------
Shirali et al., 1995	Shadow code descriptors	32	NN	97.80%
Hosseini and Bouzerdoum, 1996	Number of crossing between digit body and horizontal and vertical raster lines	10	MLP-NN	81.00%
Mowlaei et. al, 2002	Harr wavelet coefficients	64	MLP-NN	92.33%
Mowlaei and Faez, 2003	Harr wavelet coefficients	64	SVM	93.75%
Sadri et al., 2003	Derivative of 4 different views from 4 main directions using counting the number of background pixels between border and outer boundary	64	SVM	94.14%
			MLP-NN	91.25%
Soltanzadeh and Rahmati, 2004	Outer profiles of images at multiple orientation, Crossing counts, Projection histograms	257	SVM	99.57%
Mozaffari et al., 2004a	Fractal codes	64	MLP-NN	91.37%
Mozaffari et al., 2004b	Fractal codes, Harr Wavelet transform	64	SVM	92.71%
Mozaffari et al., 2005a	Fractal code	240	*k*-NN	86.30%
Mozaffari et al., 2005b	Average and variance of X and Y changes in different portion of the skeleton, …	75	*k*-NN	94.44%
Mozaffari et al., 2005c	Fractal code	240	*k*-NN, fractal transformation	92.60%
Mozaffari et al.,	Fractal code	64	SVM	92.71%

2005d				
Harifi and Aghagolzadeh, 2005	Pixels density in 12-segment digit pattern, Moment inertia, Center of mass	16	MLP-NN	97.60%
Ziaratban et al., 2007a	Position of the best occurred matching in the horizontal and vertical coordinate template, Amount of the best matching template. …	60	MLP-NN	97.65%
Alaei et al., 2009a	Chain code direction frequencies of image contour	196	SVM	98.71%
Alaei et al., 2009b	Modified chain code direction frequencies in contour. Modified horizontal and vertical transition levels	198	SVM	99.02%
Salehpour and Behrad, 2010	Automatic feature extraction using PCA	20, 30, 40, 50	SVM	95.6%
Enayatifar & Alirezanejad, 2011	Pixels accumulation, Pixels Direction	48	MLP-NN	94.30%
Mousavinasab and Bahadori, 2012	Slope variations of digit skeleton pixels	-------	k-NN	83.9%

Table 2.11 : Some Farsi handwritten letters recognition researches

Researchers	Features	No. of Features	Recognition Engine	Accuracy (Best Case)
Mowlaei et al., 2002	Harr wavelet coefficients	64	MLP-NN	91.81%
Mowlaei and Faez, 2003	Harr wavelet coefficients	64	SVM	92.44%
Alirezaee et al., 2004a	Relative energy of eroded versions respect to original image, Displacement of center of mass, ….	63	MLP-NN	97.61%
Alirezaee et al., 2004b	Invariant central moments	7	*k*-NN	90.50%
Mozaffari et al., 2004a	Fractal codes	64	MLP-NN	87.26%
Mozaffari et al., 2004b	Fractal codes, Harr Wavelet transform	64	SVM	92.00%
Mozaffari et al., 2005d	Fractal code	64	SVM	91.33%
Shanbehzadeh et al., 2007	Number of component in each character, Number and location of dots relevant to baseline, Number of pixels in each frame cell	78	Vector Quantization	87.00%
Ziaratban et al., 2008b	Terminal points, Two-way branch points, Three-way branches points	32, 40, 64, 108	MLP-NN	93.15%
Gharoie and Farajpoor, 2009	All pixels of an image	900	MLP-NN	80.00%
Alaei et al., 2010	Modified chain code direction frequencies of the contour	196	SVM	96.68%
Jenabzade et al., 2011	Wavelet coefficients from outer border, Central moments	134	MLP-NN Decision Tree	86.30%
Rajabi et al., 2012	Zoning densities, crossing count, outer profiles	315	ANN,SVM, *k*-NN, Decision Tree	97.30%
Alaei et al., 2012	Dimensional gradient	400	SVM	96.91%

Table 2.12 : Some Farsi handwritten words recognition researches

Researchers	Features	No. of Features	Recognition Engine	Accuracy (Best Case)
Dehghan et al. 2001a	Slope, curvature, number of active pixels and slope and curvature of each section extracted from the contours.	20 × number image's frames	HMM	67.18% to 96.5%
Dehghani et al. 2001	Pixels densities in various regions, contour pixels, angle of line passing through the first and end point in each image parts, ….	---------	HMM	71.82%
Safabakhsh and Adibi 2005	Moments, Fourier descriptors, Number of loops, Aspect ratio, Pixel densities, Position of right and left connections, …	9	HMM	91.00%
Broumandnia et al. 2008	Wavelet packet transform coefficients	16, 32, 96, 128, 160	*k*-NN	96.00%
Vaseghi et al. 2008	Statistical Density Values	4 × number of image frames	HMM	95.00%
Mozaffari et al. 2008b	Black – white pixel transition	10 × number of image windows	HMM	73.61%
Bagheri and Broumandnia 2009	Zernike moments	9, 25, 49, 72, 100, 182	MLP-NN	98.80%
Bahmani et al. 2010	Wavelet coefficients extracted from smoothed word image profile, …	200, 400	RBF-NN	87.60%

2.4.1 The Most Related Works in FOCR Domain

Ebrahimpor, Esmkhani and Faridi (2010) used a set of four RBF-NNs as the first stage, and another RBF-NN as the gating network in the second stage to recognize handwritten Farsi digits of Hoda dataset. The role of the last RBF-NN was assigning a competence coefficient to each initial NNs. By creating an 81-element features vector of loci features, and using only 6,000 and 2,000 samples of digits part of the Hoda dataset for training and testing the system, they finally achieved to 95.3% accuracy in the best case.

In 2011, Enayatifar and Alirezanejad proposed a FOCR system to recognize handwritten Farsi digits. They applied some pre-processing operations such as binarization, noise removal and thinning on input images, firstly. Thereafter, they divided digits into two groups ; group one including digits ‘1’, ‘2’, ‘3’, ‘4’, ‘6’ and ‘9’ and group two including digits ‘0’, ‘5’, ‘7’ and ‘8’ based on similarity on their skeletons. For feature extraction step, they divided a digit image to 24 frames, and they then calculated pixel accumulation and direction, as features, for each frame. As a result, they finally made a 48-dimentional features vector for each digit. In recognition part, they employed a MLP-NN with 50 neurons in hidden layer. They succeeded to recognize 92.70% of 20,000 testing samples of digit part of the Hoda dataset. When they decreased the number of testing samples from 20,000 to 3,000, the accuracy was improved to 94.30%.

A decision tree was used by Rajabi, Nematbakhsh and Monadjemi (2012) to recognize handwritten Persian isolated letters. They extracted 225 features for each image, including 3 different feature groups zoning, crossing count, and outer profile features. Those features were utilized in training and testing parts of their system. In classification stage, NN, SVM, and *k*-NN were employed and finally they achieved to the 97.3% accuracy in the best case. Two standard dataset IFH-CDB and Hoda were used in this research.

Alaei, Pal and Nagabhushan (2012) used eight different features sets and four different classifiers in a comparative study, for a FOCR system to recognize handwritten Farsi letters, using dataset IFHCDB. They found the gradient features, along with SVM classifier using Gaussian kernel achieves to the best result, i.e. 96.91% accuracy.

Shayegan and Aghabozorgi (2014b) used partitioning approach to reduce the volume of training dataset Hoda, to recognize handwritten Farsi digits and letters. Using 400 pixels of each input image, and a *k*-NN classifier, they achieved to 96.49% accuracy for digits, and 80.67% accuracy for characters.

2.5 Dataset Reduction

In all PR systems, the quantity, quality, and diversity of training data in the learning process directly affect the final results. In this context, the size of the training dataset is a crucial factor, because the training phase, to make system model, is often a time-consuming process. The required computational time for classifying input data increases linearly (such as in *k*-NN) or nonlinearly (such as in SVMs) with the number of samples in the training dataset (Urmanov, Bougaev & Gross, 2007). For example, time complexity for SVM grows with the square of the number of samples in the training dataset (Zhang, Suen & Bui, 2004). Hence, some of classifiers cannot be used in online or offline applications with very large number of samples in the training phase.

A major problem of PR systems is due to the large volume of training datasets including duplicate and similar training samples. However, the emergence of the Big-Data issue has caused researchers to focus their attention on data reduction in order 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 (Sanaei, Abolfazli, Gani & Buyya, 2013). 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 datasets issue reveals that two general approaches are used for dataset volume reduction: **1) Size Reduction**; and **2) Dimensionality Reduction**. The general idea behind the dataset size reduction operation is that not all the samples in the training part of a dataset have significant effect use for training the system. Therefore, if similar, duplicate, or less important samples are found, they can be removed from the training dataset, and thus, the dataset volume is reduced. 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; 1) samples far from a class centroid (outlier samples or support vector samples) (Zhongdong, Jianping, Weixin & Xinbo, 2004; Vishwanathan & Murty, 2004); 2) samples near to each class centroid (for example, using K-means clustering technique) (Vishwanathan & Murty, 2004; Ding & He, 2004). 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, and also to adjust the system parameter for better recognition.

In the dimensionality reduction technique, the system will try to find and remove the less important extracted features from existing features vectors, corresponding to dataset samples. These techniques are widely employed in different areas such as biological data clustering (Milone, Stegmayer, Kamenetzky, LóPez, & Carrari, 2013), image categorization (Benmokhtar, Delhumeau & Gosselin, 2013), heart sound signal feature reduction (Saracoglu, 2012), large time series dataset reduction (Keogh & Pazzani, 2000), automatic image segmentation (Li, Fevenes, Krzyzak & Li, 2006), EMG signal feature reduction (Phinyomark, Phukpattaranont & Limsakul, 2012), blog visualization reduction (Tsai, 2011), gene expression dataset reduction (Bronoski Borges & Nievola, 2012), face recognition (Bansal, Mehta & Arora, 2012), and so on. Specific examples of these techniques include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Random Projection (RP) (Song, Yang, Siadat & Pechenizkiy, 2013; Yang, Zhang, Zou, Hu & Qiu, 2013). However, finding an optimal, effective, and robust feature set from a big initial extracted features is usually a heuristic and difficult task (Abandah, Younis & Khedher, 2008). Sometimes, both size and dimensionality reduction techniques are used simultaneously, to reduce the volume of very large datasets (Kuri-Morales & Rogriguez-Erazo, 2009).

2.5.1 Dataset Size Reduction

There are some researches to reduce dataset size, in order to decrease the overall processing time in PR systems. Urmanov, Bougaev and Gross (2007) first calculated an original decision boundary equation *D* for each class of patterns. They then calculated the Euclidian 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 Euclidian distance between the available samples in each class 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 it is removed from the dataset. This approach, however, is very time-consuming. In addition, it is likely that by analyzing the decision boundary between class *A* and class *B*, a sample *x* of class *A* is considered as a non-important sample, and therefore, it is selected as a candidate for removal. On the other hand, by analyzing the decision boundary between class *A* and class *C*, the same sample *x* is considered as an important pattern and it should not be removed from class *A*. In these cases, it is difficult to decide whether to remove or maintain a sample.

Zhongdong, Jianping and Weixin (2004) attempted to reduce dataset volume by finding support vector samples. 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.

Vishwanathan and Murty (2004) first used SVM to categorize training system prototypes, and then found different boundaries for separating different classes of clusters. They not only removed samples which are generally close to the class boundaries, but also the typical patterns that are far from the class boundaries. They also showed a *k*-NN classifier with more diverse training samples can differentiate the input samples better than that a *k*-NN with similar training samples.

For each pair of samples from two different classes, Javed, Ayyaz and Mehmoud (2007) plotted a sphere for them, 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.

Cervantes, Li and Yu (2008) 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.

Using the subgroup discovery concept, Cano, Garcia and Herrera (2008) found a subset of a big training dataset. Each selected sample in the initial sub-dataset had special characteristics (features) in comparison to other available samples. Finally, to reduce the data volume, they used a combination of stratification and instance selection algorithms.

Shayegan and Aghabozorgi (2014b) defined a concept of similarity value and similarity interval in an OCR system. They generated a template for each class and computed a similarity value for each sample of a class to corresponding class template. They then reordered the available samples in each class based on similarities values. Thereafter, they divided the samples of each class into two partitions. The first partition included the samples which their similarity values are in closed interval [n , 1] ($n < 1$), and the second partition included the rest of the samples. Finally, they save only one sample of partition '1' and all samples of partition '2'. They succeeded to decrease the size of training part of a big dataset, without any significance decrease in recognition accuracy. The salient point of their method is it save all boundary and support vector samples in the final reduced dataset, because those samples are usually not similar to class templates. The complete details of that research are reviewed in Appendix VI.

Discussion

In summary, it is found that the majority of the aforementioned techniques can be divided into two general groups:

- The first group of techniques tries to find and delete support vector samples (SVs) in all classes (Zhongdong, Jianping & Weixin, 2004; Vishwanathan & Murty, 2004;

Cervantes, Li & Yu, 2008). These samples are usually far from the classes' centers and near to classes' boundaries. It is usual for a recognition system to classify support vector 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. Also, support vector samples are necessary to adjust system parameters in training phase. Hence, it is not a good strategy to delete all these samples from the initial dataset, in order to achieve dataset size reduction.

- 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 (Javed, Ayyaz & Mehmoud, 2007; Shayegan & Aghabozorgi, 2014b). 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.

Keeping both groups of samples, i.e. the samples near to classes' centers and the outlier samples in final reduced dataset is important goal. This is another objective of this research.

2.5.2 Dimensionality Reduction

Many different features are computed and/or extracted in the feature extraction block of a PR system. Some of the features, however, might correspond to very small details of the patterns, or that some of them are a combination of other features (non-orthogonal features), while others might not have play any effective role in the recognition stage (Shayegan & Chan, 2012). Irrelevant or redundant features may degrade the recognition results and significantly reduce the speed of 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 (Shayegan & Aghabozorgi, 2014a). Therefore, following the feature extraction process, another important process, i.e. Feature Selection (FS), is involved.

Different applications such as face recognition, license plate recognition, and image compression have used PCA technique for dimensionality reduction purpose. Also, different efforts using PCA have been made to recognize printed and handwritten characters in several languages.

Hyun-chul, Daijin and Sung Yang (2002) tried to recognize handwritten numeral from UCI dataset. They first modeled each digit class to several components, and then used PCA technique to move extracted components into a de-correlated features space. They also used the membership value method in the classification stage.

Gesualdi and Seixas (2002) employed PCA in a license plate recognition system. They used PCA for data compression in the feature extraction part, and neural networks in the recognition stage to recognize printed digits and letters that appear as strings in license plate images. They reduced the number of features from 30 to 4. They reported that the achieved accuracy for digits recognition was acceptable, but the accuracy for letters recognition was degraded significantly, when they applied PCA on data.

In 2004, Deepu, Sriganesh and Ramakrishnan employed PCA for online handwritten Tamil character recognition. The pre-processing step was carried out on a sequence of points from the digitizer and some features were extracted from each pattern. The PCA technique was then applied to reduce the dimensionality of each class. The novelty of this work lies in that it is language independent and the proposed method can be used for other scripts.

In 2005, Mozaffari, Faez and Ziaratban used fractal code as a features vector and *k*-NN classifier for handwritten Farsi zip code recognition. They applied the pre-processing operations to their dataset, and in the first step, achieved up to 86.3% accuracy. Using PCA technique, they reduced the number of fractal features to 240 and achieved 90.6% accuracy, at the end. The main disadvantage of fractal features in OCR application is computational time complexity.

To recognize handwritten Arabic isolated letters, Abandah, Younis and Khedher (2008) extracted 95 features such as image area, image width and height, image center of mass, the numbers and locations of dots in image, and so on from main body, secondary components, skeleton and boundary of each character. After that, PCA was used for feature reduction and then only the first 40 features were selected from the PCA process result. Finally, five different classifiers were employed and 87% accuracy was achieved on average in the best case.

Ebrahimi and Kabir (2008) extracted a set of 256 features in a holistic word recognition system to create a pictorial dictionary of printed Farsi words. They used the PCA technique and succeeded in reducing the number of loci features from 223 to 27.

Zhang, Suen, and Bui (2004) introduced a multi-modal approach for reducing the features dimensions in an OCR system and tested their approach on handwritten English digits dataset MNIST. They employed PCA for feature compression and succeeded in reducing the CPU time for classification.

Ziaratban, Faez and Allahveiradi (2008) proposed a novel statistical description for structure of isolated Farsi handwritten letters. They thinned the character body, decomposed a skeleton into its primitives, a curved line between any two successive feature points, and then extracted a set of features points including terminal, two-way branch and three-way branches points from primitives. Since the number of primitives varies from one character to another, they used the PCA technique to reduce and equalize the length of the features vectors. A NN-MLP with Euclidian distance was employed as classifier. To determine the exact class, they applied a post-processing stage, and achieved to 93.15% accuracy on a dataset with 11,471 samples for train and 7,647 samples for test.

Random Projection (RP) is another features selection method from group of statistical methods. RP technique is a powerful dimension reduction technique that uses random projection matrices to map the data from a huge dimensional space to a lower space. To achieve this aim, a mapping matrix R is used where the columns of matrix R are realizations of independent zero-mean normal variables, scaled to have unit length. A brief description of RP technique has been introduced in appendix V.

Shayegan, Aghabozorgi and Ram (2014) used one-dimensional and two-dimensional standard deviation and minimum to maximum spectrum diagrams to find a small subset features out of an initial features set in an OCR application. Their proposed method succeeded to decrease the dimension of features vectors to 43.6% (for Farsi digits), and to 59.4% (for English digits), meanwhile the systems accuracies were improved from 90.41% to 95.12% (for Farsi digit recognition) and from 91.93% to 94.88% (for English digits recognition). They also showed the superiority of their proposed technique compared to rival techniques PCA and RP. Table 2.13 shows related researches regarding features selection subject in OCR domain.

Table 2.13 : Summarization of researches in OCR domain, based on features selection operation

References	Feature Selection Method	System Performance				Comments
		Before Feature Selection		After Feature Selection		
		No. of Features	Accuracy	No. of Features	Accuracy	
Azmi et al., 2010	G.A.	81	77%	55	80%	
Kheyrkhah & Rahmanian, 2007	G.A.	48	75%	30	94%	FOCR-Digits
Gesualdi & Seixas, 2002	PCA	30	------	4	96%	
Deepu et al., 2004	PCA	20	89.4%	13	89.6%	Tamil characters recognition
Mozaffari et al., 2005	PCA	728-1752	86.3%	240	90.6%	Farsi zip code recognition
Abandah et al., 2008	PCA	95	84%	40	87%	Handwritten Arabic characters
Ebrahimi & Kabir, 2008	PCA	256	-------	27	99.01%	Printed Farsi words
Zhang et al., 2004	PCA	132	99.3%	10	98.4%	MNIST dataset
Bahmani et al., 2010	GA	400	86%	200	87.6%	Holistic FOCR system

Discussion

- For features selection operation, finding the correct sequence of deleting the features one by one is very important. It means that a system's derived efficiency after deleting features A, B, and C is not the same as the same system's derived efficiency after deleting the features in order A, C, and B or B, C, and A and so on

(El-Glaly & Quek, 2011). Due to their nature, some features are relevant to others from different view of points. In this case, SBS techniques do not help to find the best subset of features. For example, El-Glaly and Quek (2011) extracted four feature sets S1, S2, S3 and S4 (with some common features) to use in an Arabic OCR system. They trained the system with these four feature sets, separately. After that, these sets were delivered to a PCA algorithm, and PCA rearranged the features based on their importance in the recognition system. The results showed that feature X in rank 23 in set S3 took rank 7 in set S1 and so on. This experiment indicates that if feature X is deleted for the sake of feature reduction, it may cause a large error in final results.

- The main problem concerning GA methods is that they always select chromosomes one by one with the best recognition percentage, and move this chromosome (feature) to the next stage. However, it is possible that when a good characteristic feature gets combined to another feature, the overall performance will not be as good as the individual performances. Also this technique is very time consuming.
- Based on the FOCR literature, the PCA technique has produced better results in comparison with the other two categories of feature selection techniques. However, although PCA has been widely utilized in different PR applications, it suffers from high computational cost. Computation of PCA requires eigenvalue decomposition of the covariance matrix of the features vectors with around $O(d^3 + d^2n)$ computations, where n is the number of samples and d is the dimensionality of the features space (Sharma & Palimal, 2008). Hence, this powerful technique is usually employed for features extraction/selection operation on small scale datasets. Also, choosing the

optimum number of features, based on Eigen vectors, from the new re-ordered features space, created by PCA, is not an easy task.

CHAPTER 3

RESEARCH METHODOLGY

3.1 Introduction

This chapter explains the used research methodology in the study. The sub-topics include the description of the employed methods, to achieve the research objectives, mentioned in Chapter 1. Also, the approach for evaluation of the proposed model is presented. The Hoda dataset (Khosravi & Kabir, 2007) for handwritten Farsi digits and letters, and also MNIST dataset (Le Cun, Bottou, Bengio & Haffiner, 1995) for handwritten English digits, which are used in this research for evaluating proposed techniques, will be introduced.

3.2 Approaches to Research

Figure 3.1 demonstrates the research methodology framework of this thesis. Each stage is explained in the following sub-sections, briefly.

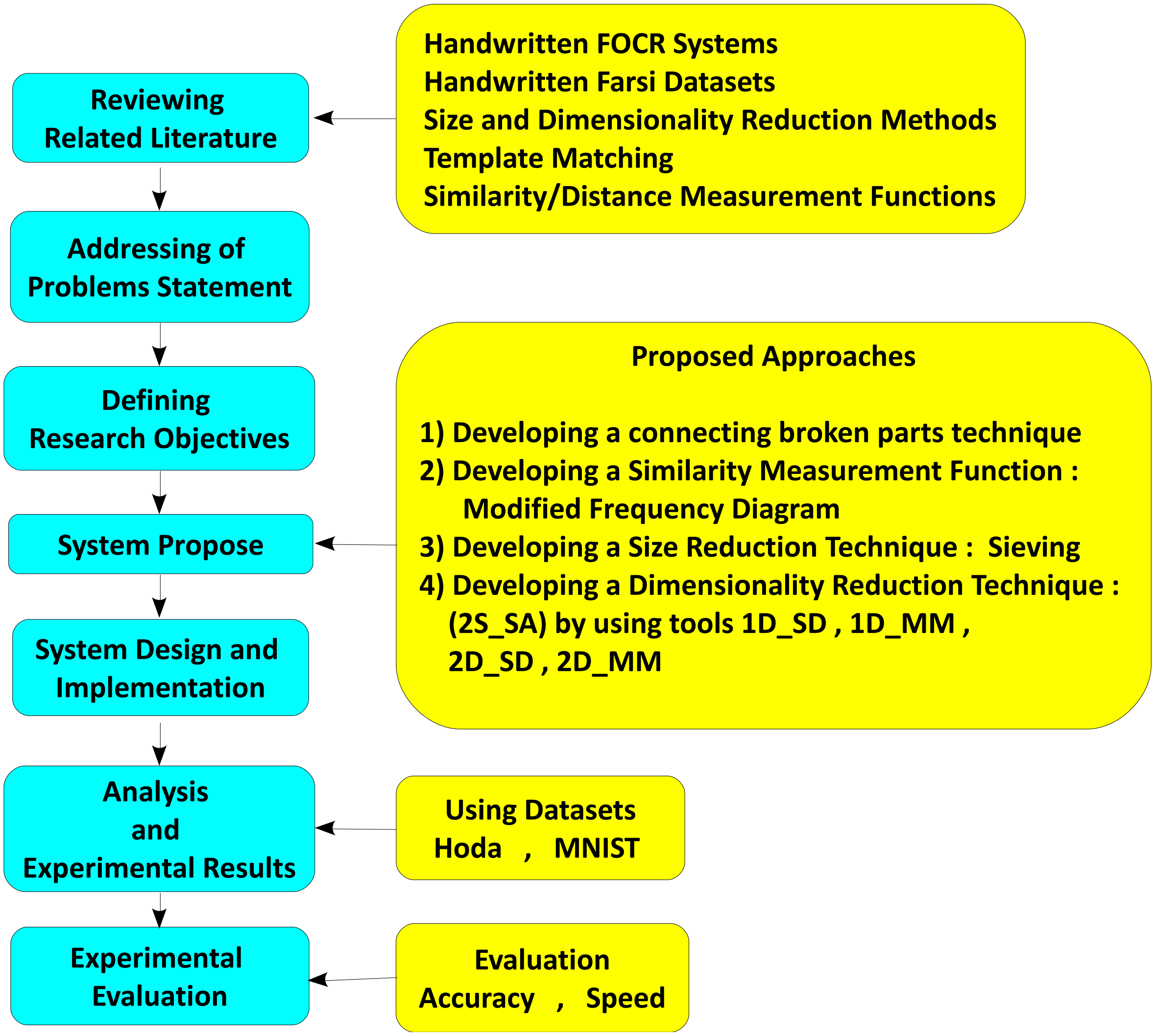

Reviewing
Related Literature
Handwritten FOCR Systems
Handwritten Farsi Datasets
Size and Dimensionality Reduction Methods
Template Matching
Similarity/Distance Measurement Functions
Addressing of
Problems Statement
Defining
Research Objectives
Proposed Approaches
1) Developing a connecting broken parts technique
2) Developing a Similarity Measurement Function :
Modified Frequency Diagram
3) Developing a Size Reduction Technique : Sieving
4) Developing a Dimensionality Reduction Technique :
(2S_SA) by using tools 1D_SD , 1D_MM ,
2D_SD , 2D_MM
System Propose
System Design and
Implementation
Analysis
and
Experimental Results
Using Datasets
Hoda , MNIST
Experimental
Evaluation
Evaluation
Accuracy , Speed

Figure 3.1 : Research methodology framework

3.2.1 Reviewing Related Works

Based on reviewing the literature, handwritten FOCR systems were introduced (Section 2.1.1). The characteristics of Farsi writing and features of Farsi alphabet were addressed (Section 2.2 and Figure 2.3). The existing FOCR systems for Farsi digits and alone mode of Farsi characters were discussed (Section 2.4, Table 2.10, Table 2.11, Table 2.12). Also, different techniques for size reduction (Section 2.5.1) and dimensionality reduction (Section 2.3.3.2, Section 2.5.2) in OCR systems were discussed. The analyzing of the existing reduction methods gives an accurate view of advantages and disadvantages of them. It had been stated that pre-processing, feature extraction and selection, and recognition phases are essential parts in any FOCR systems. Various similarity/distance measurement functions for template matching operation are introduced in Appendix IV and section 4.4.3 and equation 4.5.

3.2.2 Problem Formulation

The literature review on handwritten FOCR systems clearly indicated that in spite of the valuable advancements in designing FOCR systems in recent years, but their performance is still far from humans, both in terms of accuracy and speed (Table 2.10, Table 2.11, Table 2.12). Also, the literature showed the available size and dimensionality reduction techniques do not produce satisfaction results for FOCR datasets (section 2.5).

3.2.3 Identifying Research Objectives

Formulation of problems provides direction for this research to come up with the following objectives:

1) To enhance the output quality of pre-processing blocks in a handwritten FOCR system.
2) To propose a new technique for dataset size reduction, in a handwritten FOCR system, to speed up system training and testing.
3) To propose a new technique for dimensionality reduction, in a handwritten FOCR system, to speed up system training and testing, and increasing the system accuracy.
4) To test and evaluate the capability of the proposed methods in improving the performance of a FOCR application, by applying it on Farsi digits and letters.

To achieve these objectives, a model is proposed and evaluated in this study. The following sub-section explains briefly about the proposed model.

3.2.4 Proposed Model for FOCR Systems

To achieve the first objective (**to enhance the quality of output pre-processing module**), some common pre-processing operations such as noise removal filtering (Section 2.3.2.2), size normalization, translation, slant correction (Section 2.3.2.4), and the new technique **C**onnecting **B**roken **P**arts (**CBP**), in order to connect broken parts of an image, are applied on input images. The detailed description of the new technique CBP is described in section 4.3.3.

To address the second objective (**dataset size reduction**), it had been stated that it is necessary to keep both groups of samples near to class centers and outlier samples in final reduced training datasets. Here, in order to increase the overall system speed, a template for each class is created, and sieving operation is applied on training datasets, by using the new dataset size reduction method; **S**ieving **B**ased **R**eduction (SBR) technique (Section 4.4). Figure 3.2 depicts the general structure of the proposed dataset size reduction method SBR. The complete description of the proposed method is identified in Section 4.4.4. The results of the mentioned size reduction method will be reported in Table 4.2 and Table 4.3, and Section 5.2.

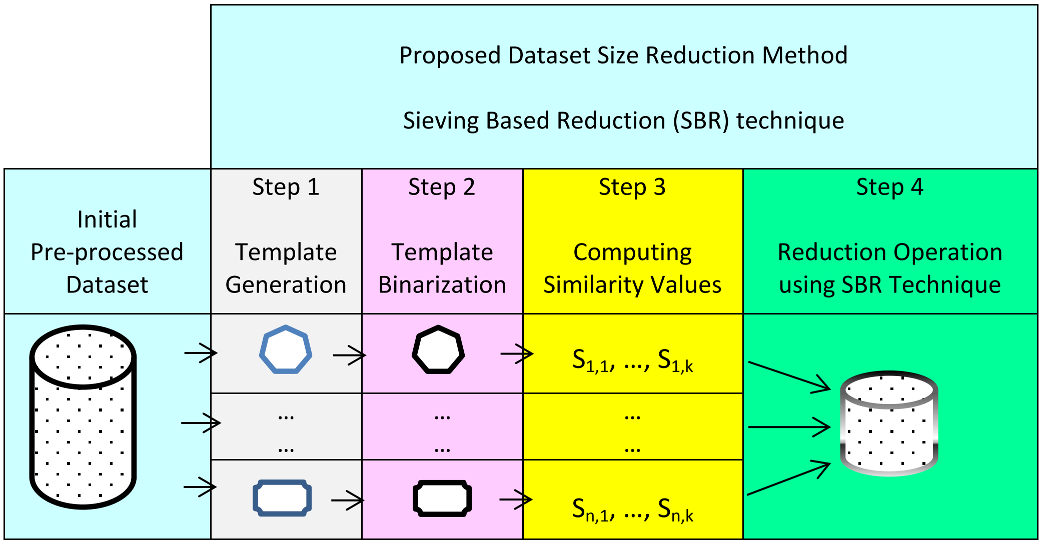

Proposed Dataset Size Reduction Method
Sieving Based Reduction (SBR) technique
Initial Pre-processed Dataset
Step 1
Template Generation
Step 2
Template Binarization
Step 3
Computing Similarity Values
Step 4
Reduction Operation using SBR Technique
$S_{1,1}$, …, $S_{1,k}$
…
…
$S_{n,1}$, …, $S_{n,k}$

Figure 3.2 : The proposed model for dataset size reduction

To achieve the third objective (**dimensionality reduction**), the new technique **2-S**tages **S**pectrums **A**nalysis (**2S-SA**) is applied on initial features vectors. To such an aim, One-Dimensional Standard Deviation (1D_SD) spectrum diagrams (Section 4.6.1.1), Two-Dimensional Standard Deviation (2D_SD) spectrum diagrams (Section 4.6.1.2), One-Dimensional Minimum to Maximum (1D_MM) spectrum diagrams (Section 4.6.1.1), and Two-Dimensional Minimum to Maximum (2D_MM) spectrum diagrams analysis (Section 4.6.1.2) are introduced to make the final reduced features vectors (Section 4.6.2).

Figure 3.3 depicts the overall structure of the proposed dimensionality reduction model. The model receives a features vector including the most-used features, based on the literature, as the initial features vector denoted as ***Initial_S***. Dimension reduction is taken in two stages. In the first stage, using the proposed tools 1D_SD and 1D_MM spectrums analysis, the number of features is decreased from ***n***, in the ***Initial_S***, to ***k***, in the first reduced version of features vector, i.e. ***S1.*** Then, in the second stage, and by employing the proposed tools 2D_SD and 2D_MM spectrums analysis, the number of features is decreased again from ***k***, in the features vector ***S1***, to ***p***, in the final reduced version of features vector, i.e. ***S2***. Unlike the other available techniques for dimensionality reduction, such as PCA, the proposed method can keep every feature in the final features vector, based on some characteristic of a specific feature, or even based on user opinion. The complete description of the proposed method is identified in Section 4.6.2. The results of the mentioned feature selection method will be reported in Table 4.6, and Section 5.3.

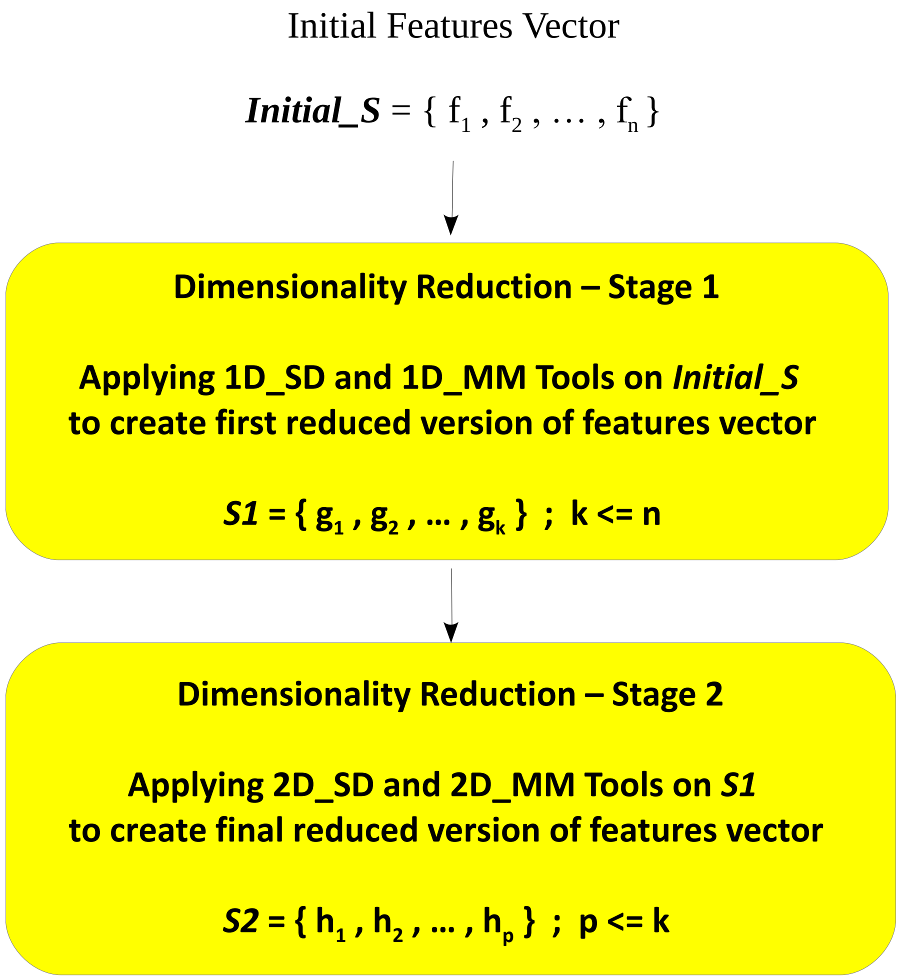

Initial Features Vector
Initial_S = { f_1 , f_2 , … , f_n }
Dimensionality Reduction – Stage 1
Applying 1D_SD and 1D_MM Tools on *Initial_S*
to create first reduced version of features vector
S1 = { g_1 , g_2 , … , g_k } ; k <= n
Dimensionality Reduction – Stage 2
Applying 2D_SD and 2D_MM Tools on *S1*
to create final reduced version of features vector
S2 = { h_1 , h_2 , … , h_p } ; p <= k

Figure 3.3 : The two-stage proposed model for dimensionality reduction

Figure 3.4 shows a general model for an OCR system. It includes six main modules: data acquisition, pre-processing, segmentation, features extraction, recognition (classification), and post-processing. The sixth module, post-processing, is not mandatory part, and sometimes it is taken into consideration to improve the final results. The model does not include the segmentation module, when the separate letters and digits are manipulated (similar to this research), or when a holistic approach without segmentation process is used to recognize a limited vocabulary.

Figure 3.5 demonstrates our proposed system architecture for an offline handwritten FOCR system. In the proposed model, the input image goes through the steps pre-processing, data size reduction, features extraction, dimensionality reduction, and recognition. Based on the reason that the scope of this research is recognition of digits and alone letters, hence the proposed model does not include the segmentation module.

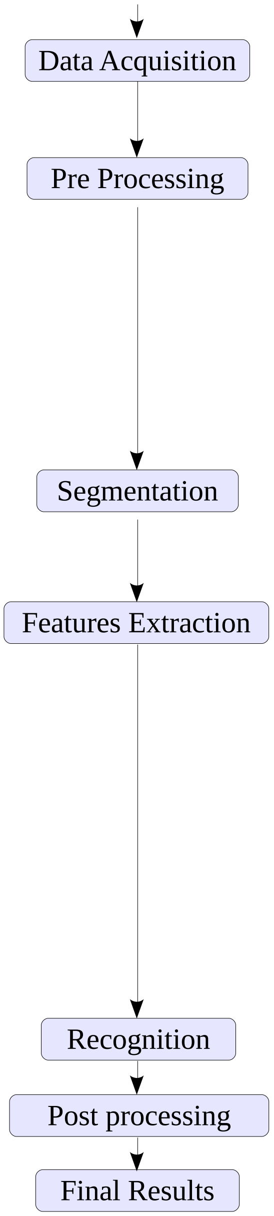


Figure 3.4

General model for an OCR system

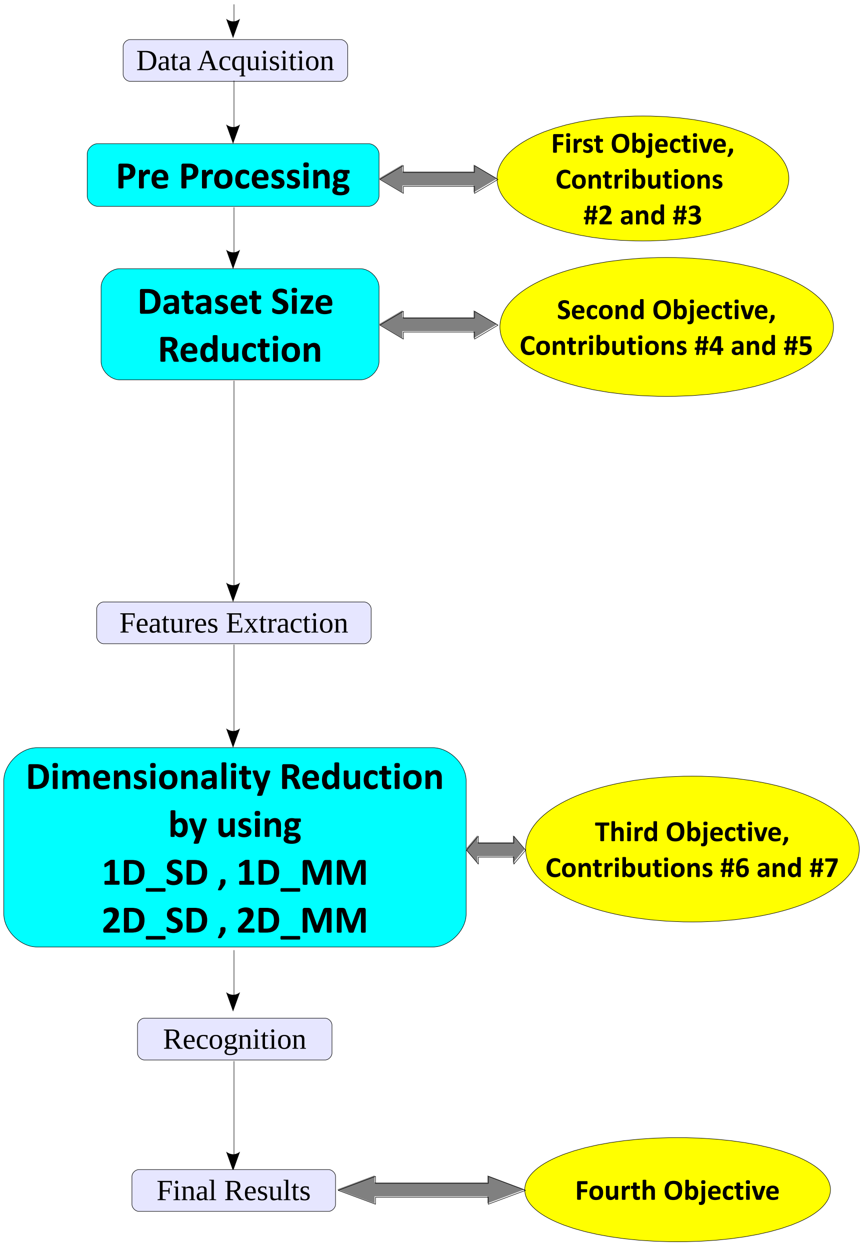

Data Acquisition
Pre Processing
First Objective, Contributions #2 and #3
Dataset Size Reduction
Second Objective, Contributions #4 and #5
Features Extraction
Dimensionality Reduction by using 1D_SD , 1D_MM 2D_SD , 2D_MM
Third Objective, Contributions #6 and #7
Recognition
Final Results
Fourth Objective

Figure 3.5

The proposed model for a FOCR system

3.3 Choosing Datasets for Experiments

To evaluate a proposed technique in a real OCR application, some datasets are needed. In OCR domain, scanners, cameras, cell phones, and so on are common devices for image acquisition process. Usually, scanners are used for generating standard datasets for OCR applications. The images of both of datasets Hoda and MNIST, used in this thesis, have been generated using scanners. The first one, Hoda dataset, includes handwritten Farsi digits and also separate letters images. This standard benchmark dataset have been used frequently for handwritten FOCR research (Alaei, Pal & Nagabhushan, 2009; Ebrahimpor, Esmkhani & Faridi, 2010; Enayatifar & Alirezanejad, 2011; Rajabi, Nematbakhsh & Monadjemi, 2012; Rashnodi, Sajedi & Saniee 2011). The second one, MNIST dataset, includes handwritten English digits which it has been employed in various researches in English OCR domain (Le, Duong & Tran, 2013; Impedovo, Mangini & Barbuzzi, 2014; Feng, Ren, Zhang & Suen, 2014; De Stefano, Fontanella, Marrocco & Scotto di Freca, 2014).

3.3.1 Hoda Dataset

The Hoda dataset is one of the largest handwritten Farsi standard datasets. This dataset was created in 2007 by Khosravi and Kabir, and it includes two sections: digits, and letters. The digit section was prepared by extracting the images of Farsi digits from 11,942 application forms related to university entrance exams. 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. Similar to other languages, there are 10 digits ‘0’ to ‘9’ in the Farsi Alphabet. However, in handwritten Farsi manuscripts, digits ‘2’, ‘4’, ‘5’ and ‘6’ are written in two different forms. Digit ‘2’ is written as ‘٢’ or ‘r’, digit ‘4’ is written as ‘٤’ or ‘۴’ , digit ‘5’ is written as ‘٥’ or ‘۵’, and digit ‘6’ is written as ‘۶’ or ‘٦’. The digits section of the Hoda dataset contains 80,000 samples, and has been divided into two parts, namely 60,000 training samples and 20,000 testing samples. It is a balanced dataset that includes 6,000 and 2,000 samples for each digit in the training and testing parts, respectively. Figure 3.6 shows some sample digits from this dataset.

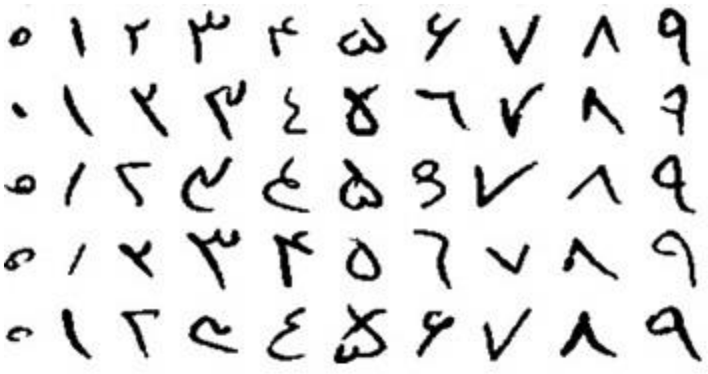


Figure 3.6 : Some samples of digits part of the Hoda dataset

Characters part of the Hoda dataset was prepared also from the mentioned forms. There are 32 individual and four extra letters in Farsi alphabet that they made all alone mode of Farsi alphabet. Figure 3.7 shows a sample of alone mode letters from characters part of the Hoda dataset.

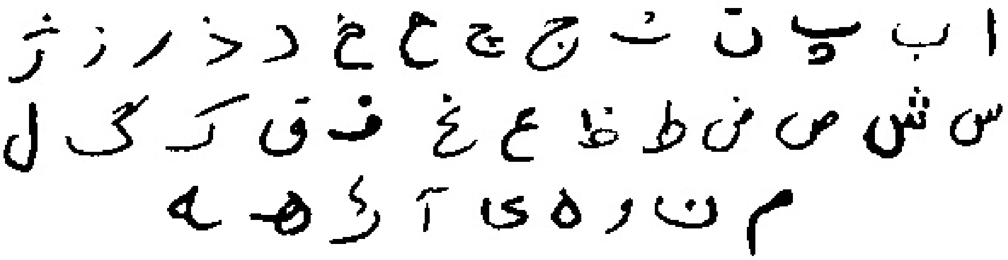


Figure 3.7 : Some samples of characters part of the Hoda dataset

Similar to digits part, characters part in the Hoda dataset is divided into two parts: training and testing. The character part of the Hoda dataset is not balance, because different classes have different number of samples. Numbers of samples in the training and testing part for each letter have been shown in Table 3.1. The total numbers of samples in characters part of the Hoda dataset are 70,645 and 17,706 for training and testing part, respectively (www.ocr.irnbsdvjbjdsfbgvakjdfgkjdf).

Table 3.1: Number of samples in the Hoda dataset – characters part

Label	Persian Spell	Character	Number of Train Samples	Number of Test Samples	Total Count
0	Alef	ا	2,080	520	2,600
1	Be	ب	2,070	519	2,589
2	Pe	پ	2,084	519	2,603
3	Te	ت	2,070	520	2,590
4	The	ث	1,843	462	2,305
5	Jim	ج	2,075	519	2,594
6	Che	چ	1,978	496	2,474
7	He	ح	2,082	521	2,603
8	Khe	خ	2,072	519	2,591
9	Dal	د	2,080	520	2,600
10	Zal	ذ	1,830	460	2,290
11	Re	ر	2,078	520	2,598
12	Ze	ز	2,072	520	2,592
13	Zhe	ژ	2,071	519	2,590
14	Sin	س	2,067	520	2,587
15	Shin	ش	2,075	520	2,595
16	Sad	ص	2,073	520	2,593
17	Zad	ض	2,078	520	2,598
18	Ta	ط	2,076	518	2,594
19	Za	ظ	1,924	481	2,405
20	Ain	ع	2,082	520	2,602
21	Ghain	غ	2,072	520	2,592
22	Fe	ف	2,063	519	2,582
23	Ghe	ق	2,066	519	2,585
24	Kaf	ک	2,071	520	2,591
25	Gaf	گ	2,075	520	2,595
26	Lam	ل	2,072	520	2,592
27	Mim	م	2,081	520	2,601
28	Noon	ن	2,076	520	2,596
29	Vav	و	2,077	520	2,597
30	Ha (single)	ه	2,079	520	2,599
31	Ya	ی	2,078	520	2,598
32	Hamze	ئـ	1,989	507	2,496
33	Alef-Hat	آ	641	161	802
34	Ha (binocular)	ـهـ OR هـ	1,889	473	2,362
35	Ha (sticky end)	ـه	456	114	570
		Total Count	**70,645**	**17,706**	**88,351**

3.3.2 MNIST Dataset

The Modified National Institute of Standards and Technology (MNIST) dataset contains 60,000 training and 10,000 testing samples (http://yann.lecun.com/exdb/mnist/index.html). This dataset is an unbalanced dataset (Table 3.2). All the digits have been stored in 28x28 dimension pixels, with intensities from 0 to 255. Figure 3.8 shows some sample digits from this dataset.

Table 3.2 : Distribution of digits in the MNIST and Hoda datasets

Digit	**MNIST**		**Hoda**	
	Train	**Test**	**Train**	**Test**
0	5,923	980	6,000	2,000
1	6,742	1,135	6,000	2,000
2	5,958	1,032	6,000	2,000
3	6,131	1,010	6,000	2,000
4	5,842	982	6,000	2,000
5	5,421	892	6,000	2,000
6	5,918	958	6,000	2,000
7	6,265	1,028	6,000	2,000
8	5,851	974	6,000	2,000
9	5,949	1,009	6,000	2,000
Total	**60,000**	**10,000**	**60,000**	**20,000**

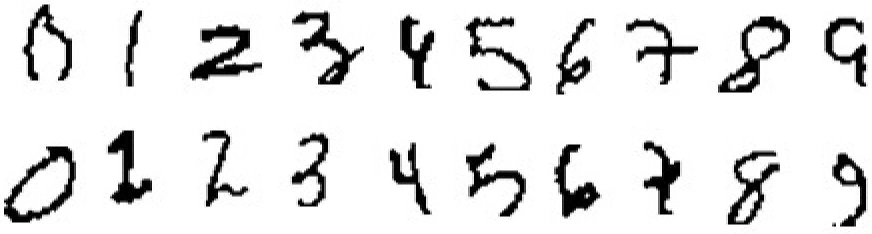


Figure 3.8 : Some digit samples of the MNIST dataset

3.4 Evaluation Process and Cross Validation

Any classification model should be evaluated to find its advantages and disadvantages. Recognition accuracy is the most common parameter for this aim. Accuracy estimation refers to the process of approximating the future performance of the model to classify new unknown samples (Kohavi, 1995).

Usually, the available samples in a dataset are categorized into two separate parts: training and testing. However, when there is only one small dataset, the Cross Validation (CV) technique is used to create more training and testing parts. The process of CV is a way of getting more reliable estimate out of possibly scarce data. It divides the whole of dataset randomly into k number of equal size separate folds. This is called *k*-fold CV technique. K-1 folds are used for training and the remained fold is used for testing the system. This process is repeated for k times, and for each of iterations, one new fold is used in testing part. In this way, each sample is guaranteed to be in the test set once. The CV technique causes that all samples contribute in both training and testing operations. Finally, the mean of *k* time recognition accuracy is reported as final accuracy.

One disadvantage of using CV technique is the elapsed time for k iteration training and testing processes. Hence, the k-fold CV with a high value for k may be infeasible in practice. Kohavi (1995) showed the best choice for k is 10. In this research, three big handwritten alphanumeric standard dataset: 1) digits part of the Hoda dataset including 60,000 training samples and 20,000 testing sample; 2) characters part of the Hoda dataset including 70,645 training samples and 17,706 testing sample; and 3) the MNIST dataset including 60,000 training samples and 10,000 testing sample were used for testing recognition purpose. These dataset are big datasets, hence, the researchers usually do not use them in CV mode. However, in some cases of dimensionality reduction step, the 10-fold CV technique (Kohavi, 1995) is used to do better assessment about recognition accuracy.

3.5 Summary

This chapter introduced the methodology adopted and different techniques which have been employed in this thesis. Hence, according to research objectives, a research methodology framework was proposed. A model was proposed for FOCR systems in this study. The model involved some new items: a technique for connecting broken parts of an image in order to increase the accuracy, a method for dataset size reduction in order to speed up the system operation, and a technique for dimensionality reduction in order to speed up the system operation and increase the accuracy. The details of each part of the model are explained in the next chapter. Finally, the evaluation plan for the proposed model was explained.

CHAPTER 4

THE PROPOSED MODEL FOR DATA REDUCTION IN HANDWRITTEN FOCR SYSTEMS: DESIGN, IMPLEMENTATION, AND EXPERIMENTS

4.1 Introduction

In this chapter, pre-processing, dataset size reduction, and dimensionality reduction modules, of the proposed model for a FOCR system, are explained in details. However, the other parts of the proposed system, which have been used in common ways, are demonstrated briefly. All the proposed methods are applied on selected datasets and the results are reported. The comparison between the proposed methods results and related woks in the domain is postponed to Chapter 5.

In the proposed model, in order to enhance the quality of input images (Section 1.5, Objective #1), input data are firstly pre-processed (Section 4.3). Then, a new dataset size reduction operation (Section 4.4) is applied on datasets in order to decrease the volume of the training data (Section 1.5, Objective #2). This reduction operation not only saves the memory usage in practice, but also decrease the processing time in different parts of the system, especially in the training phase which different classifiers NNs, *k*-NN, or SVM are used. Most of the OCR systems utilize conventional methods to generate a features vector for recognition task. In this thesis, this conventional approach is followed (Section 4.5) along with a new dimensionality reduction method (Section 4.6) in order to speeding up the system (Section 1.5, Objective #3). In the final stage, i.e. recognition phase, *k*-NN, NN, or SVM classifiers are employed (Section 1.5, Objective #4). The detailed comparison between the proposed methods and the literature are reported in Chapter 5.

4.2 Overview of the Proposed Model for an Offline Handwritten FOCR System

An offline FOCR system, like to other OCR systems for other languages, includes some common stages. The first step is image acquiring process which enters different images into the system. The input images may be in color, gray scale or binary formats. However, if the format of an image is not binary, the system will convert it. Then, pre-processing step is usually used to enhance the quality of input images. Next step includes important features extraction and features selection operations. The following step is recognition operation, as a heart of any PR systems.

The general structure of the proposed model was shown in Figure 3.5. It included the pre-processing, dataset size reduction, features extraction, dimensionality reduction, and recognition parts. Figure 4.1 depicts these steps, briefly, to address objectives ‘1’ to ‘4’ of this study.

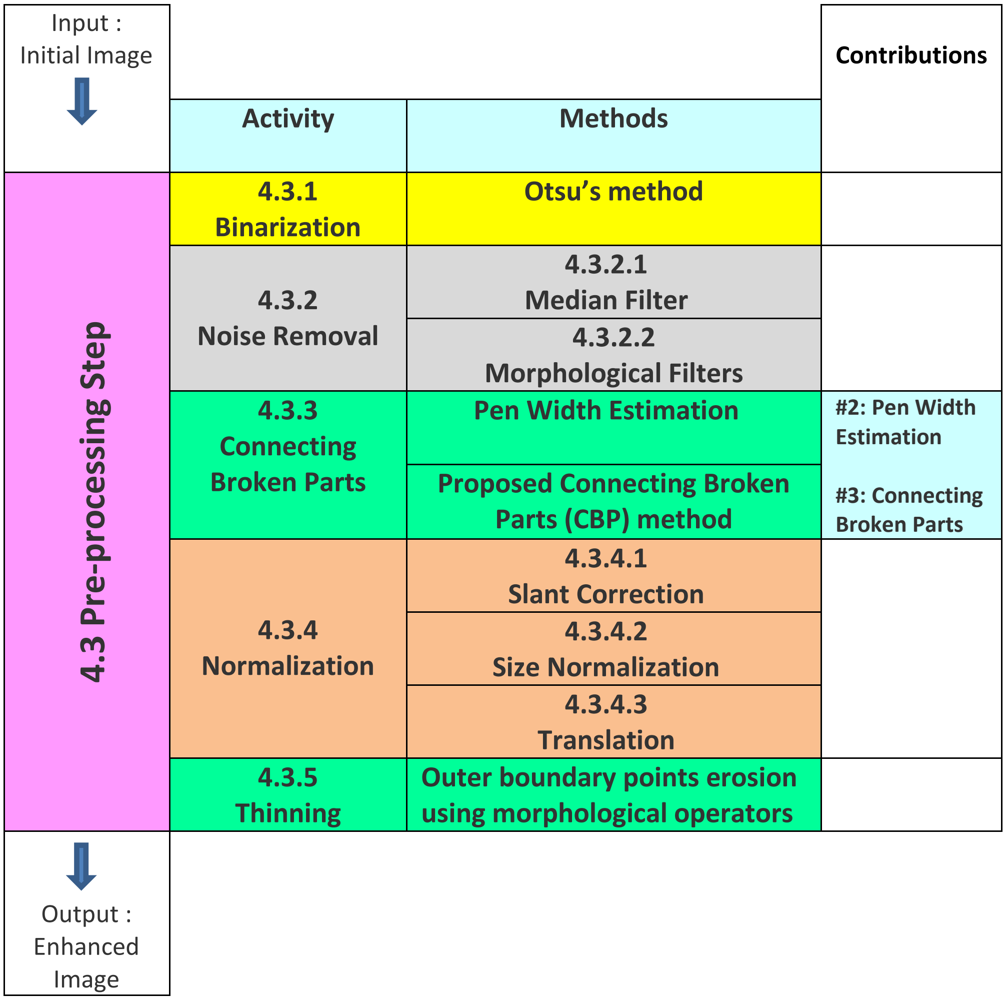
Steps	Activities	Contributions	Objectives
4.3 Pre-processing	4.3.1 Binarization		#1 Enhancing the quality of output of pre-processing step
	4.3.2 Noise Removal		
	4.3.3 CBP Method	#2 : Pen Width Estimation #3 : Connecting Broken Parts	
	4.3.4 Normalization	------------	
	4.3.5 Thinning		
4.4 Dataset Size Reduction	4.4.1 Template Generation	------------	#2 Proposing a new technique for dataset size reduction in order to speed up system training and testing
	4.4.2 Template Binarization		
	4.4.3 Computing Similarity Value	#4: New Similarity Measurement Function	
	4.4.4 Reduction Operation	#5: Sieving Method SBR	
4.5 Features Extraction	Feature Extraction based on the Literature	------------	------------
4.6 Dimensionality Reduction	4.6.1.1 1D_SD and 1D_MM	#6: Dimensionality reduction method 2S_SA #7: Suggesting a Small Features Set for FOCR Systems	#3 Proposing a new technique for dimensionality reduction in order to speed up system training and testing and increasing the accuracy
	4.6.1.2 2D_SD and 2D_MM		
4.7 Recognition	*k*-NN, NN, SVM	------------	#4 System test and evaluation

Figure 4.1 : Overview of the proposed FOCR system

4.3 Pre-processing

In this research, an extra pre-processing operation is carried out to address the first objective, i.e. **Enhancing the quality of pre-processing module output**. The contributions of this section are: ***1) Proposing a formula to estimate the pen width, effectively, in a handwritten manuscript; 2) Proposing a new method to connect the broken parts of an image together.***

Pre-processing operations play an important role in different image processing applications. In the case of OCR applications, the final performance depends very much upon the quality of the original data. Hence, the important pre-processing operations, such as translation, noise removal, dimension normalization, slant correction, and so on by using common powerful techniques are first performed on the input images. The outputs of this stage are some clean and enhanced images that they are suitable for recognition system. Figure 4.2 shows an overview of the available parts in the pre-processing step of the proposed model.



Input : Initial Image

	Activity	Methods	Contributions
4.3 Pre-processing Step	4.3.1 Binarization	Otsu’s method	
	4.3.2 Noise Removal	4.3.2.1 Median Filter	
		4.3.2.2 Morphological Filters	
	4.3.3 Connecting Broken Parts	Pen Width Estimation	#2: Pen Width Estimation #3: Connecting Broken Parts
		Proposed Connecting Broken Parts (CBP) method	
	4.3.4 Normalization	4.3.4.1 Slant Correction	
		4.3.4.2 Size Normalization	
		4.3.4.3 Translation	
	4.3.5 Thinning	Outer boundary points erosion using morphological operators	

Output : Enhanced Image

Figure 4.2 : Overview of the pre-processing operations

4.3.1 Binarization

Entered images to an OCR system may be in color, gray scale or binary format. However, most of OCR softwares operate on binary images. Hence, colored and gray scale images should be converted to binary format. Transforming a colored image to gray-level format is usually carried out by Equation 2.1. After that, a gray level image is converted to binary version by using one of the available approaches. Complementary details have been described in Section 2.3.2.1.

All available samples in the Hoda dataset (Section 3.3.1) are in binary format, while the available samples in the MNIST dataset (Section 3.3.2) have been stored in gray level format. Hence, in this step, by analysing the input images grey level histograms, and by using the standard global Otsu's method (Section 2.3.2.1), the MNIST samples are changed to bi-level images.

4.3.2 Noise Removal

Different sources of noises may affect the initial images in any OCR systems. Low quality of initial images, low quality of papers, low performance of scanners, low resolution of initial scanned images, lack of accuracy in writing by writers, and so on are some examples of noise sources. It is necessary to remove these noises to achieve the better system accuracy.

In this research, two common well known noise removal methods, i.e. median filtering (Section 2.3.2.2.a) and morphological filtering (Section 2.3.2.2.b), are utilized to reduce the noise level.

4.3.2.1 Median Filter

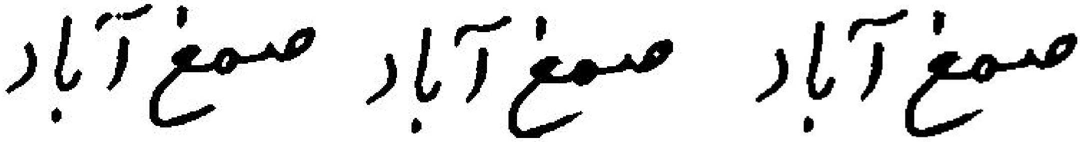
First noise removal method includes a median filter with a 3×3 window (Equation 2.2, Section 2.3.2.2.a). It is applied to remove salt and pepper noise from the images. Median filter reduces the contrast of images and also removes some small holes, but it preserves the structural shape of the images. Figure 4.3 shows an example for applying median filter on an image of Farsi letter ‘گ’.

a) Farsi letter 'گ' with 'salt and pepper' noise b) The image after noise removal operation

Figure 4.3 : Applying median filter for noise removal

4.3.2.2 Morphological Filters

Second noise removal method includes morphological operation with using opening and closing operators. These operations apply dilation and erosion morphological operations on image using 2×2 unity structural element (Equations 2.3 to 2.6) to remove high-frequency noise from the input images. These operators not only delete some noisy pixels from the image, but also stick some disconnect parts of the images. Figure 4.4 demonstrates an example for applying opening and closing process on a Farsi word. In this example, a specific structural element matrix 2×2 with all members '1' has been used.



a) Initial image b) Image after applying closing operator c) Image after applying opening operator

Figure 4.4 : Shape of Farsi word 'صمغ آباد' from dataset IAU/PHCN

4.3.3 Connecting the Broken Parts of a Sample (CBP method)

Although, the morphological closing operator connects near small islands in an image, but if the broken parts are far from each other, closing operator cannot connect them together. Figure 4.5 indicates an example for this case which some breaks have been rebuilt, but some breaks are still available.

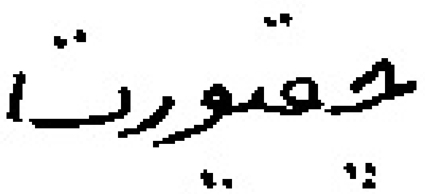


Figure 4.5 : A sample word after applying morphological closing operator, with some available breaks

The body of all Farsi and English digits is constructed using only one component (Figure 3.6, Figure 3.8). Thus, after applying closing operation on an input image, if there is still more than one group of connected pixels in a pre-processed digit input image, the extra parts are considered as noise or separate components of the initial image. To find and remove the rest of the noise from an input image, the **P**en **W**idth (**PW**) is estimated firstly by using the biggest component of the image. PW estimation is carried out by using three different methods, and then the average of these values is considered to be the final pen width as follows:

E_1 = The mode of image vertical projection
E_2 = (The value of image density) / (The number of image skeleton pixels)
E_3 = {(The value of image density) / (The number of image outer profile pixels)} × 2

$$\text{PW} = \frac{E_1 + E_2 + E_3}{3} \tag{4.1}$$

The experimental results showed that the average of three aforementioned estimated values is a more accurate estimate of the PW than each of them, alone. After finding the PW value, all small components with a pixel density less than two times of the PW value are considered as noise, and deleted from the input image. The threshold '2' was obtained experimentally for datasets Hoda and MNIST. The rest of the connected components are considered as broken parts of the digit image.

To connect the broken parts of a digit image, the new proposed **C**onnecting **B**roken **P**arts (**CBP**) method is suggested and applied on the digits parts of the MNIST and Hoda datasets.

By using connected component analysis, the biggest available part of an image is found and named as the main part $\boldsymbol{M}$. The outer contour of the main part $\boldsymbol{M}$ is then extracted and the coordinates of its pixels are saved in array ***MAIN***. Thereafter, for all of the rest secondary components $\boldsymbol{S_i}$ (which they are smaller than the main part $\boldsymbol{M}$), the outer contour is found and saved the pixels coordinate of those outer contours in another array ***SEC***. Then, the Euclidean distance is computed between all elements of array ***MAIN*** with all elements of array ***SEC.*** The smallest value of the computed distance indicates the shortest path between contour $\boldsymbol{M}$ and one of the secondary contours $\boldsymbol{S_k}$. Finally, a line with thickness equal to estimated PW value is drawn along the shortest path between $\boldsymbol{M}$ and $\boldsymbol{S_k}$. As a result, the main part $\boldsymbol{M}$ is connected to a secondary part $\boldsymbol{S_k}$. This process was repeated until there is not another secondary component. In each round, a new version of main part $\boldsymbol{M}$ is used, because one secondary part is connected to the old version of the main part. Algorithm 1 demonstrates the pseudo code for this process.

Algorithm 1. The Connecting Broken Parts (**CBP**) method.

Input: image **I** with one main part, **M,** and **k** secondary parts

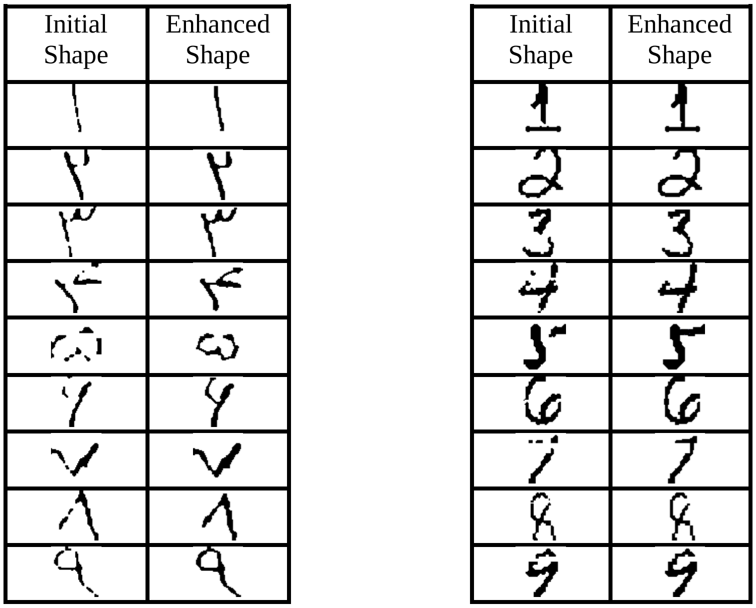
k: number of secondary components

PW : Pen width value /* computed by using Equation 4.1 */

Output: image **I’** with only one main part

```
while (k <> 0) do   %There is another secondary component in input image
    clear SEC;
    Initialize h = 1 ;
    MAIN = The pixels coordinates of outer contour of the main part M;
    repeat
        t = The pixels coordinates of outer contour of image secondary part $S_h$ ;
        append t to array SEC;
        h++ ;          /* increment h */
     until ( h <= k)  %there is not another secondary parts in image);
     for (each pixel A in array MAIN)
    {
           for (each pixel B in array SEC)
             {
                    d = distance (A , 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 PW thickness from A_min to B_min) ;
      k -- ;   % Decreasing the number of secondary components
end while;
return (I’)
```

Figures 4.6.a and 4.6.b show examples of applying CBP method on two digits sets of the Hoda and the MNIST dataset, in order.



(a) Farsi digits (b) English digits

Figure 4.6 : Applying the new proposed CBP method on input images

The aforementioned proposed CBP method is a time consuming process, but the final accuracy of a PR system is completely dependent to quality of input data. Hence, this algorithm is applied on the input images, in addition to other pre-processing operations, to increase the quality of input data. The CBP method produced very good results in this research, but in some cases it failed, too. Figure 4.7 shows one of the available images of digit ‘8’ in MNIST dataset that the proposed CBP method could not connect two end point pixels together. Here, the image has a break in its body, but it includes only one main part, without any secondary parts. Hence, the CBP method could not find any extra parts to connect them to the digit’s main body.

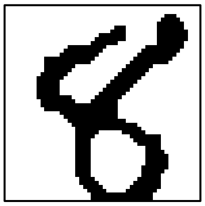


Figure 4.7 : The weakness of CBP method in order to connect two end points

It should be noted that the CBP method is not applied for Farsi letters, because the majority of Farsi letters includes more than one part in their bodies, i.e. secondary part such as dot(s), slanted bar(s), Hat bar, and Hamze (For example: 'گ', 'پ', 'آ', 'ت', and so on), which they should not be connected to the main body of letter.

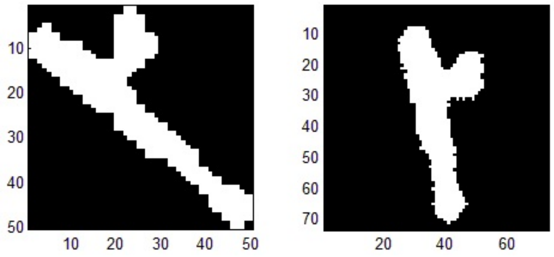
4.3.4 Normalization

Handwritten digits and letters are written in very vast styles and sizes. In order to apply different techniques on available patterns, all input samples are put in a standard form in size, translation, and rotation.

4.3.4.1 Slant Correction

Usually in handwritten documents, writers follow their own style in writing process. One of the measurable factors of different handwriting styles is word slant (Section 2.3.2.4.a). To achieve better performance in feature extraction and then recognition steps, it is better that the slant angle is detected and corrected.

In order to find and correct the slant angle of each image, the Hanmandlu's method (Section 2.3.2.4.a) is used. In practice, it was found if this operation is repeated two times, the results will be more accurate. Figure 4.8 shows a sample of digit '۲' ('2'), from the Hoda dataset, before and after slant correction.

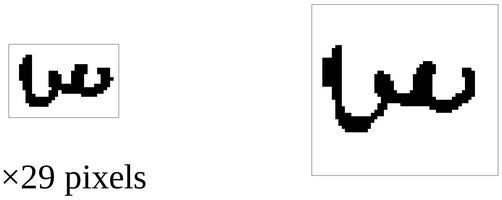


a) Initial Image b) Image after slant correction

Figure 4.8 : Farsi Digit ‘۲’ (‘2’) before and after slant correction

4.3.4.2 Size Normalization

In this step, by considering the initial aspect ratio of each sample, the size of any image is changed to 50×50 pixels. However, the initial width and height of the most samples are not equal. Hence, zero padding operation is applied in the most cases for one of image's dimensions. Also, a 20×20 version of input image is created to use elsewhere which there is a need to smaller images. Figure 4.9 shows the images of Farsi letter ‘س’ from the Hoda dataset before and after size normalization.



16×29 pixels

50×50 pixels

a) Initial image b) Image after size normalization

Figure 4.9 : An image sample of Farsi letter ’س‘ before and after size normalization

4.3.4.3 Translation

In this step, by shifting the center of mass of any image to the center of 50×50 pixels window (Section 2.3.2.4.c), normalization operation respect to translation is carried out.

4.3.5 Thinning (Skeletonization)

Some important features for alphanumeric characters, such as end, branch, and crossing points, are extracted from skeleton of the images. Hence, thinning operation (Section 2.3.2.5) by using Cranny's method is applied on an enhanced binary image to create a skeleton of it. Cranny's thinning method produces a suitable skeleton that preserves the patterns' original forms, and it does not create bogus data (Gonzalez, Woods & Eddins, 2009). Figure 4.10 shows Farsi digit '۴' ('4') and its skeleton.

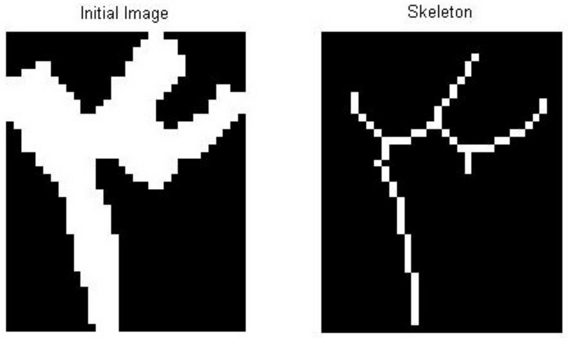


Figure 4.10 : Farsi digit '۴' ('4') and its skeleton

4.3.6 Experimental Results for Pre-processing Operations

To show the impact of applying the proposed pre-processing operation CBP, some experiments were carried out to recognize testing samples of MNIST dataset and digits part of the Hoda dataset. In the first step, the input samples, via applying only traditional pre-processing operations, without CBP method, were used. In the second step, the input samples, via applying traditional pre-processing operations and also CBP method, were used.

In the first experiment, no pre-processing operations were applied on input images. For Farsi digits recognition, a MLP-NN was trained with 133 neurons (Table 4.6) in the input layer (corresponding to the number of elements in feature vector ***Initial_S***), 30 neurons (found experimentally) in the hidden layer, and 10 neurons (corresponding to 10 different classes of digits ‘0’ to ‘9’) in the output layer, respectively. The network was trained with all 60,000 samples from the training part, and was then tested with all 20,000 samples from the testing part of the Hoda dataset. Finally, the system achieved to 47.19% accuracy. In the second experiment, the system was trained with the pre-processed version of samples, without applying the proposed algorithm CBP. On average, the correct recognition rate increased from 47.19% to 83.77%, clearly indicating the impact of pre-processing operation on accuracy. Finally, in addition to apply traditional pre-processing operations (noise removal, slant correction, scaling, and translation), the proposed CBP method was applied on input images. Here, the accuracy was again improved from 83.77% to 90.4%, clearly showing the impact of CBP pre-processing operation on final accuracy.

The aforementioned experiments were repeated to recognize handwritten English digits of MNIST dataset. Three versions of samples, i.e. initial raw samples, pre-processed samples by using traditional techniques, and pre-processed samples by using traditional operation and CBP technique, were used. Here, the network included 133 neurons in input layer, 30 neurons (found experimentally) in the hidden layer, and 10 neurons (corresponding to 10 different classes of available English digits) in the output layer, respectively. Also, the network was trained with all 60,000 samples from the training part, and was then tested with all 10,000 samples from the testing part of the dataset. In this case, the accuracy was increased from 55.85% to 86.08%, and from 86.08% to final accuracy 91.93%, clearly indicating the positive impact of pre-processing operations on English digits recognition. The achieved accuracies, before and after applying pre-processing operations, are reported in Table 4.1.

Table 4.1 : The impact of the proposed CBP pre-processing operations on recognition accuracy

	Accuracy			
Dataset	Without Pre-processing	Applying Traditional pre-processing (Noise Removal, Normalization)	Applying Traditional pre-processing + the proposed CBP method	Improvement by using CBP method
Hoda, Digits Part	47.19%	83.77%	90.4%	6.63%
MNIST	55.85%	86.08%	91.93%	5.85%

4.4 The Proposed Method SBR for Dataset Size Reduction

In this research, the dataset size reduction operation is carried out to address the second objective, ***i.e. reducing the volume of training dataset in order to speeding up the system training and testing.*** The contributions of this section are: ***1) Proposing a new similarity measurement function; 2) Proposing the new sieving method for size reduction.***

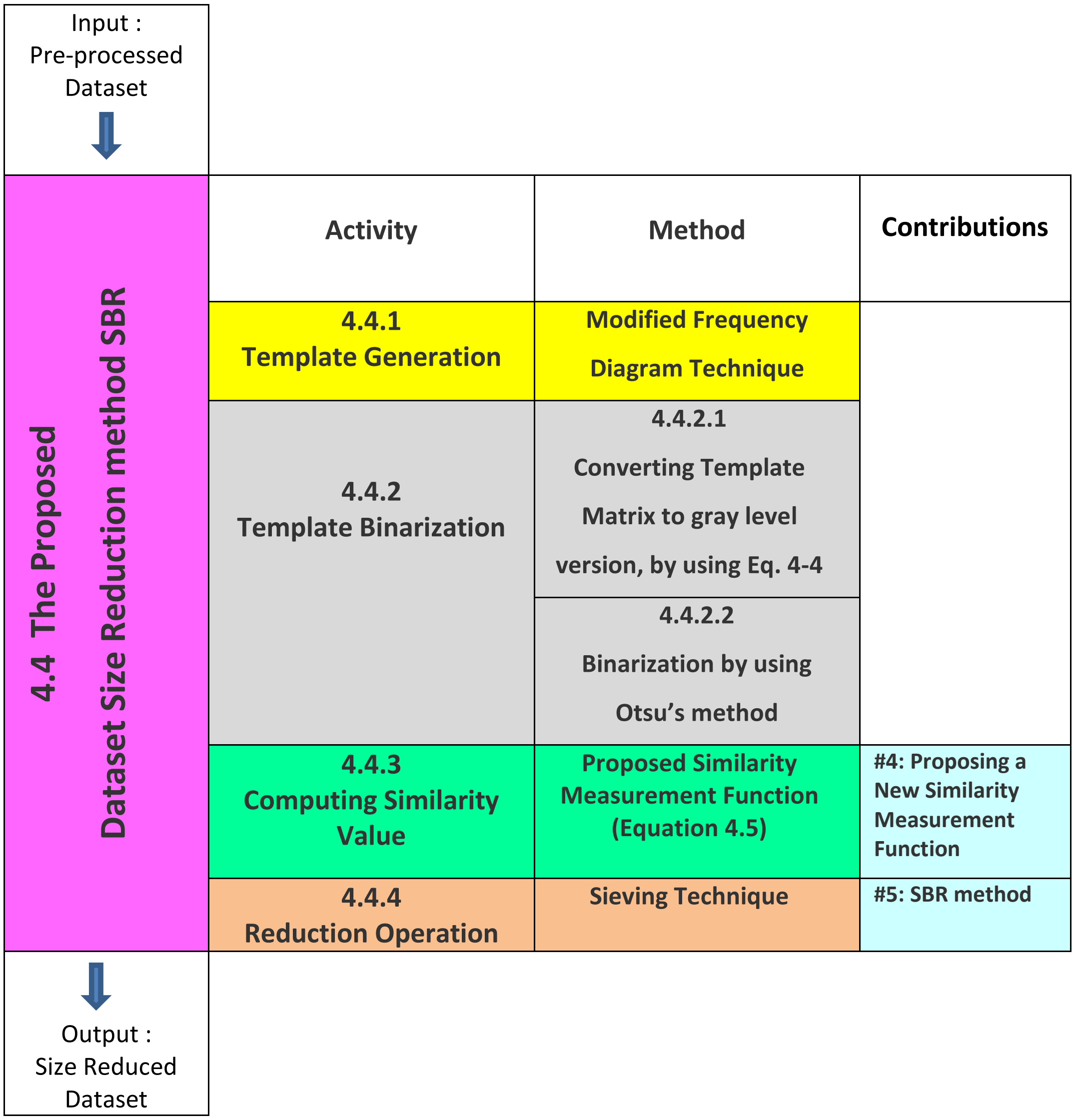
As it was mentioned before in Section 2.5, PR systems often have to handle the problem of large volume of training datasets including duplicate and similar training samples. This problem leads to large memory requirement for saving and processing data, and the time complexity for training algorithms. Hence, reducing the volume of training part of a dataset, in order to increase the system speed, with minimum decrease in system accuracy, can be considered as a good goal in PR systems.

The main idea behind the dataset size reduction operation is that in a large dataset, some of the training samples in a class might fully or nearly resemble each other, and therefore, they cannot produce different crucial information to the training stage of a PR system. Therefore, if the (almost) similar samples in a class are found, then it is possible to keep only some of them and remove the rest from dataset. As a result, the volume of a dataset can be decreased.

A survey of the literature on large dataset issue (Section 2.5) reveals that two general approaches have been used for dataset size reduction: 1) finding and removing samples far from a class centroid (outlier samples, support vector samples, boundary points); and 2) samples near to the centers of classes. However, the outlier and support vector samples are important samples to evaluate the system efficiency and also adjusting the system parameters. Also, the samples near to a class centroid include important information about the class characteristics, and they are necessary to make an accurate system model.

In the most of available methods for dataset size reduction, researchers compare all samples from a class to all samples in another class to find the boundary points between all class pairs. Finally, they put these samples in final dataset as important key samples (or delete them as unimportant samples in making system model). It is obvious that this approach need to compare all samples of a class with all samples of other classes, that it is very time consuming process. Also, it needs a huge memory for manipulating these volumes of data.

In this thesis, a new method is presented to reduce the number of samples in the training section of a dataset. The proposed method utilizes a completely different approach in compare to the other available methods. Here, the samples of a class are compared to other samples in the same class, based on their similarities to that class template. The proposed algorithm for reducing the number of training dataset samples for an OCR system (and generally for some other pattern recognition systems which they used pictorial datasets) has four main parts including: 1: Template generation for each class, 2: Template binarization, 3: Calculating similarity value between each instance and corresponding class template, and 4: Deleting some patterns from each class and reducing the class size, by using sieving operation (Figure 4.11). The initial input to this module is the output of pre-processing module (Section 4.3). The steps of the proposed dataset size reduction method are demonstrated in the following sub-sections.



Input :
Pre-processed
Dataset

4.4 The Proposed
Dataset Size Reduction method SBR

Activity	Method	Contributions
4.4.1 Template Generation	Modified Frequency Diagram Technique	
4.4.2 Template Binarization	4.4.2.1 Converting Template Matrix to gray level version, by using Eq. 4-4	
	4.4.2.2 Binarization by using Otsu's method	
4.4.3 Computing Similarity Value	Proposed Similarity Measurement Function (Equation 4.5)	#4: Proposing a New Similarity Measurement Function
4.4.4 Reduction Operation	Sieving Technique	#5: SBR method

Output :
Size Reduced
Dataset

Figure 4.11 : Overview of the proposed dataset size reduction module

4.4.1 Template Generation for each Class

In template matching techniques, it is necessary to generate a template for each class to compare every sample with corresponding class template and find the similarity/distance value between a sample and a template. In this research, Frequency Diagram (FD) is used in order to make a template for each class (Downton, Kabir & Guillevic, 1988). FD is defined as the number of occurrences of pixel '1' in the coordinates (x,y) for all available samples in a specific class, as follows:

$$D_i\,(x,y) \;=\; \sum_{n=1}^{N_i} \big(F_n(x,y)\big) \tag{4.2}$$

x , y : coordinate of different pixels of any samples
x = 1, …, k ; y = 1, …, p (k, p : Dimensions of normalized sample)
n : *n*'th sample of a specified class
$F_n(x,y)$: pixel value of *n*'th sample at coordinate (x,y)
N_i : The total numbers of samples in class C_i

In Modified version of Frequency Diagram (MFD) (Khosravi & Kabir, 2007), the pixel density variable D_i is increased by one unit, if the pixel in coordinate (x,y) of an input sample is ‘1’, and the pixels density variable D_i is decreased by one unit, if the pixel in coordinate (x,y) of an input sample is ‘0’ (Equation 4.3). Indeed, MFD creates a more accurate description of pixels density in each class.

$$D_i\,(x,y) \;=\; \sum_{n=1}^{N_i} (F_n(x,y) * 2 - 1\,) \tag{4.3}$$

Figures 4.12 and 4.13 show examples of FD and MFD for Farsi digit ‘٧’ (‘7’) obtained by using Equations 4.2 and 4.3, respectively. For simplicity, only 200 samples of digit ‘7’ were used for generating these two figures.

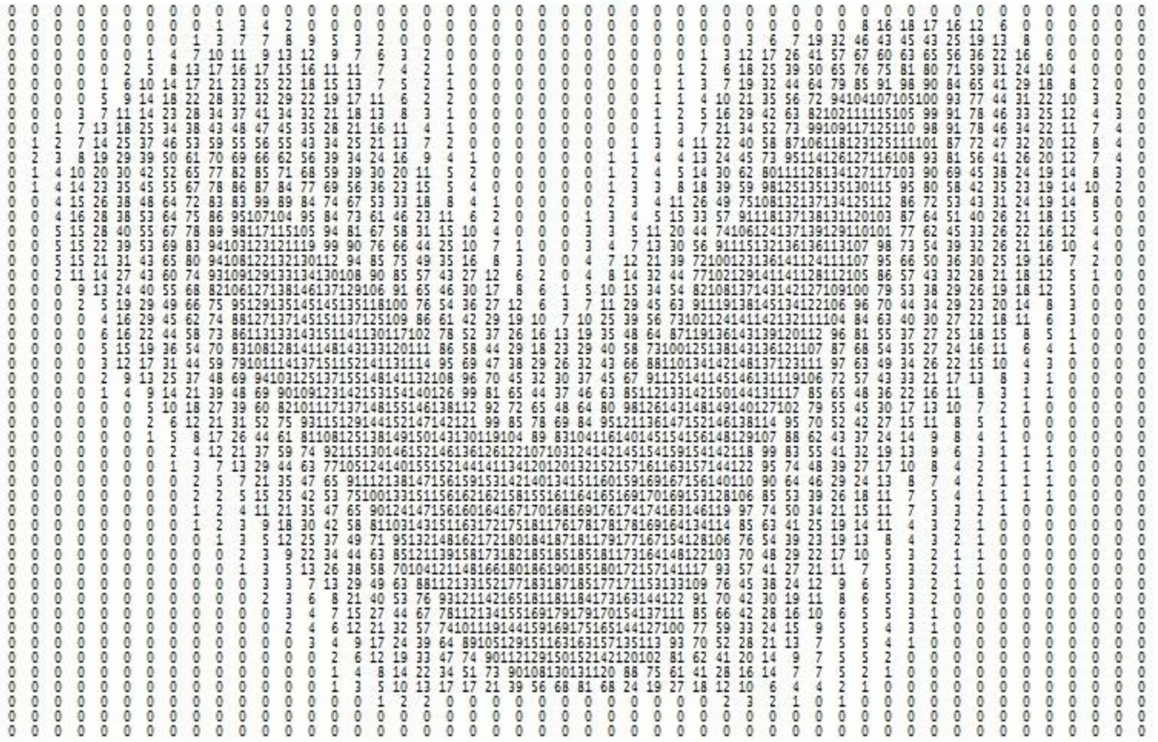


Figure 4.12 : Frequency Diagram (FD) matrix for
Farsi digit '۷' ('7') using Equation 4.2

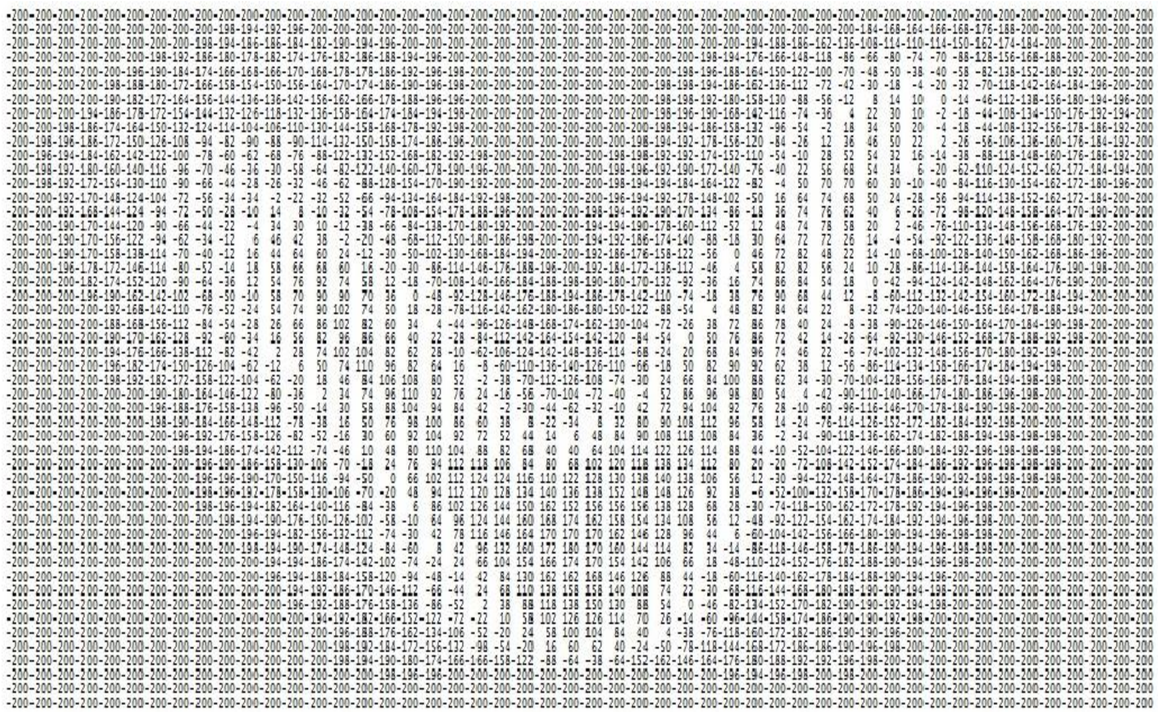


Figure 4.13 : Modified Frequency Diagram (MFD) matrix for
Farsi digit '۷' ('7') using Equation 4.3

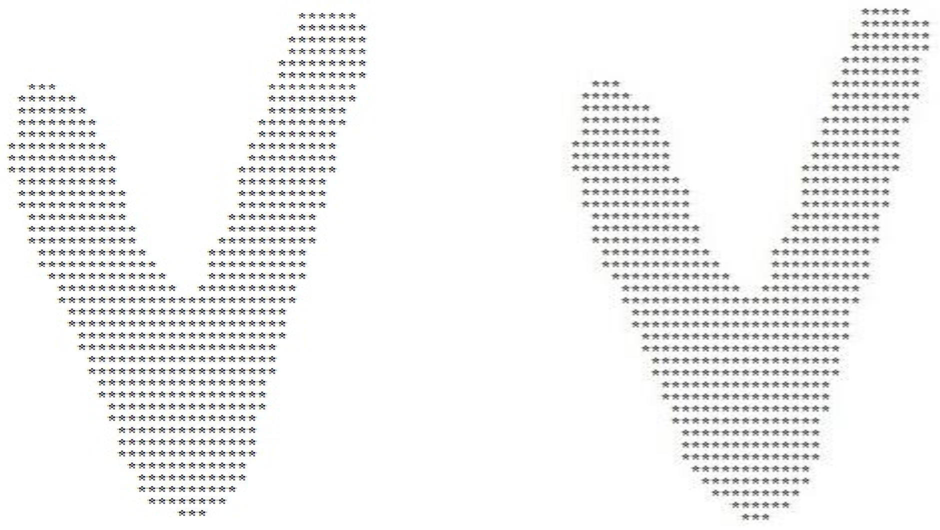
The obtained values for FD, by using Equation 4.2, are the numbers from 0 to N_i where N_i is the total numbers of samples in class C_i. These values can be normalized into range 0 to 100 by multiplying the results in 100/N_i. Also, the obtained values for MFD, by using Equation 4.3, are the numbers from -N_i to N_i. They can be normalized into range (-100) to (100) by multiplying the results in 100/N_i. In this thesis, the MFD version (Equation 4.3) is used, because the MFD concept provides a more accurate description for similarity/distance between an image and a class template. The MFD matrices are considered as templates for different classes, called Template Matrices (TMs).

4.4.2 Template Binarization

In order to compare the samples of a class with the derived template for the same class, a TM is converted to Binarized Template Matrix (BTM) version. This operation involves two steps. In the first step, the values of the TM elements are scaled to the gray levels spectrum from 0 to 255 by using Equation 4.4, 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.

$$\text{GL\_TM} = \frac{(P + N_i)}{2} * \frac{255}{N_i} \tag{4.4}$$

In the second step, the GL_TM elements are converted to the BTM version by using the standard global Otsu's method (Section 2.3.2.1). Figures 4.14.a and 4.14.b show binarized template related to obtained templates in Figures 4.12 and 4.13 using the mentioned method, in order.



a) BT corresponding to Figure 4.12 b) BT corresponding to Figure 4.13

Figure 4.14: Binarized Template (BT) for Farsi digit '۷' ('7')

4.4.3 Computing Similarity Value

Template matching techniques involves computing similarities between an input instance and generated classes' templates, by using a similarity value measurement function. Based on this reason that a class template is generated by counting similar pixels '1' and '0' among all samples in a class, hence it is better to use a similarity function which it includes the concepts of frequency diagram matching.

By using the MFD Equation 4.3, a similarity variable **S** is defined. Similarity variable $\mathbf{S}_{\mathbf{k,i}}$ indicates the similarity value between the *k*'th sample of class *i* to 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, $\mathbf{S}_{\mathbf{k,i}}$ is decreased. It means that the employed similarity measurement function should use the 'Exclusive NOR' operator to show similarity and 'Exclusive OR' to show non-similarity between the image pixels and template pixels. Also, in order to amplify the effect of similar pixels in comparison to non-similar pixels, the effect of the equal pixels was considered to be twice that of non-equal pixels. The general equation of this new concept is defined as follows (while the ***w*** reward coefficient is set to 2, experimentally):

$$S_{k,i} = \sum_{x=1}^{n} \sum_{y=1}^{m} \{w * [f_{k,i}(x,y) \odot BTM_i(x,y)] - [f_{k,i}(x,y) \oplus BTM_i(x,y)]\} \, |MFD_i(x,y)| \tag{4.5}$$

i : class number in pattern space
n , m : image dimensions
w : reward coefficient (in this research, this parameter was set to 2)
$f_{k,i}$: k'th sample image of class i
BTM_i : i'th Binarized Template Matrix (corresponding to class C_i)
MFD_i : i'th Modified Frequency Diagram matrix (corresponding to class C_i)
$\odot$: Logical XNOR operator
$\oplus$: Logical XOR operator

Experimental results showed that the calculated values for similarity variables $S_{k,i}$, proposed in Equation 4.5, have wide variances, and this leads to better differentiation between the samples in a class. The ***w*** reward coefficient in Equation 4.5 was set 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 a too big value for ***w*** will cancel the penalty effect related to the corresponding non-similar pixels in Equation 4.5. Appendix IV describes some other existing similarity/distance measurement functions, briefly.

4.4.4 Reduction Operation using Sieving Approach

After generating a template for a class (Section 4.4.1), the class template is changed to binary version (Section 4.4.2), and then, by Equation 4.5, the similarity values between all samples in a class and corresponding template are computed. Thereafter, the sieving operation is performed for dataset size reduction purpose.

Sieving-Based Reduction (SBR) Method: In this method, the proposed similarity measurement function Equation 4.5 is used to find a **S**imilarity **V**alue (SV) between a sample and its class template. This process is repeated for all training samples. Finally, all the training samples in a particular class are sorted in descending order based on their computed SVs. By performing sampling operation at the rates of 1/2, 1/3, and 1/4, using the sorted version of the training dataset, three versions of reduced training dataset are finally kept: half; one-third; and one-fourth. For example, the digits part of the original dataset Hoda contains 60,000 training samples. Hence, three reduced version of this dataset, including 30,000 samples (half version), 20,000 samples (one-third version), and 15,000 samples (one-fourth version) are created. These three reduced versions of the training dataset, as well the initial training dataset, are used in system training and testing. Figure 4.15 depicts the overall process for this method. Also, algorithm 2 explains the proposed SBR method for dataset size reduction.

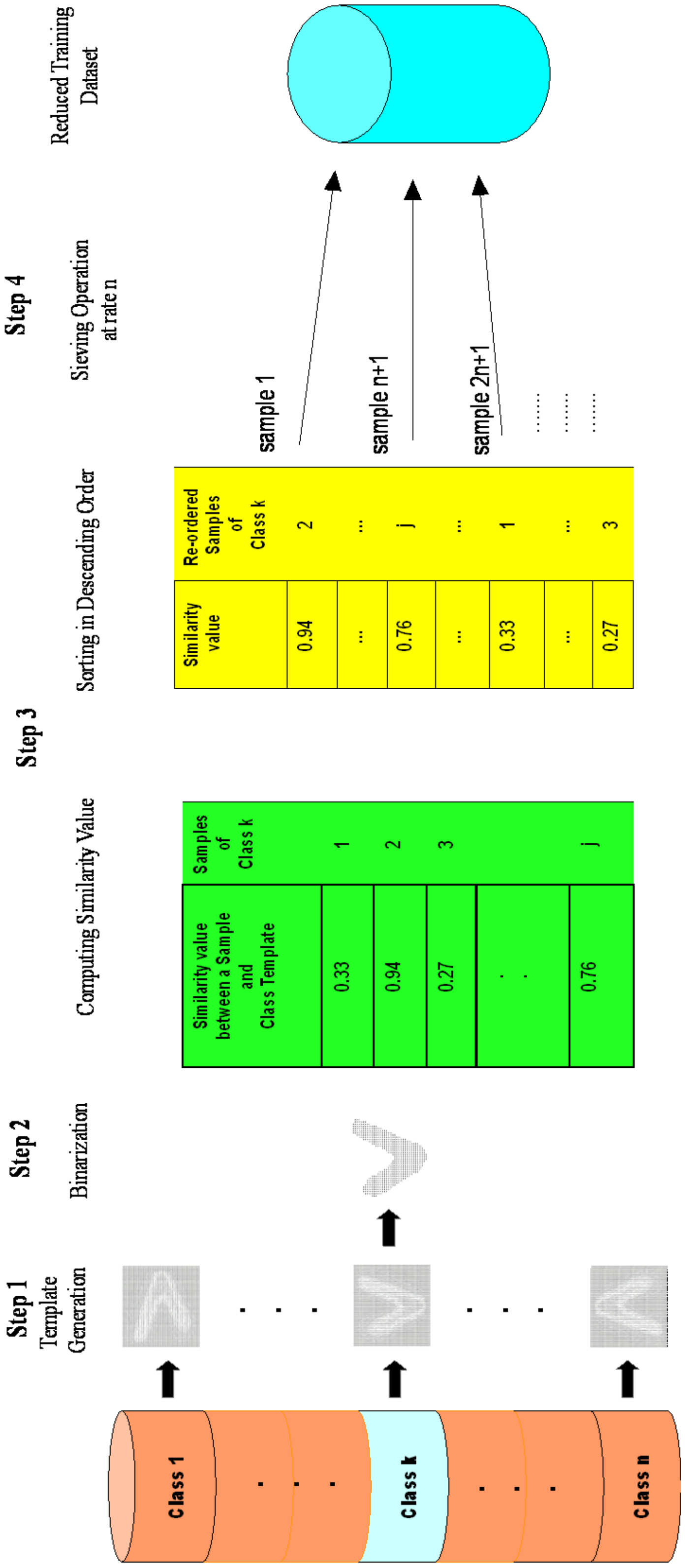


Figure 4.15 : The flow diagram for the proposed dataset size reduction method SBR

Algorithm 2. SBR method for dataset size reduction purpose

Input:

IPTD : Initial Pre-processed Training Dataset /* ***IPTD*** = $\{f_1, f_2, \dots f_n\}$ */
k : Sampling Rate
noc : Number of Classes in Pattern Space

Variables:

CLASS : A matrix including all sample of a class in class space
***T* :** A template matrix corresponding to a class in class space
GL_T : Gray level matrix corresponding to matrix ***T***
BT : Binary version of matrix ***GL_T***
SV : Similarity value vector corresponding to samples a class
n : Number of training samples in **IPTD**

Output:

***R* :** Final reduced version of dataset with m samples (m < n)

Method: SBR

```
R := null ;
for i = 1 : noc do
{
      CLASS = IPTD (Class(i)) ;
      /* Using Modified Frequency Diagram Equation 4.3, and all the j
         samples of matrix CLASS */
      T = Template_Generation_Function (CLASS) ;
      /* Using Equation 4.4 */
      GL_T = Gray_ Level_Generator_Function (T) ;
      /* Using Otsu's method */
      BT = Binary_ Matrix_Generation_Function (GL_T) ;
      /* Using similarity measurement function 4-5 */
      for k=1 : number of samples in CLASS do
      {
            SV(k) = Similarity_Value_Measurement_Function (BT , CLASS(k)) ;
      }
      Sorted_Class = Sort (SV) ;    /* Descending order sort */
      for j = 1 : n by step k do
      {
            Insert Sorted_Class(j) to R;
      }
}
return (R) ;
```

4.4.5 Experimental Results using the Proposed Dataset Size Reduction Method SBR

Various experiments were performed on Farsi Hoda dataset, to investigate the effect of the proposed dataset size reduction method SBR on recognition accuracy. The following subsections demonstrate the experiments.

4.4.5.1 Experiments on Digits Part of the Hoda Dataset

By sampling operation at the rates 1/2, 1/3, and 1/4, three reduced training dataset versions were finally kept for Farsi digits: half (30,000 samples) by using sampling rate 1/2, one-third (20,000 samples) by using sampling rate 1/3, and one-fourth (15,000 samples) by using sampling rate 1/4. The original training dataset with 60,000 samples and the mentioned three reduced versions were used as training datasets for a FOCR system. All the 400 pixels of an image were fed directly to a *k*-NN classifier, as input information. The final results are shown in Figure 4.16 and Table 4.2. The results clearly shows that the recognition speed increased to more than double, while the accuracy decreased only 0.68%, slightly from 96.49% to 95.81%. When the reduced 1/3 version of the training dataset was used, the recognition speed changed to more than 3 times faster, but the accuracy dropped by 1.42% from 96.49% to 95.07%.

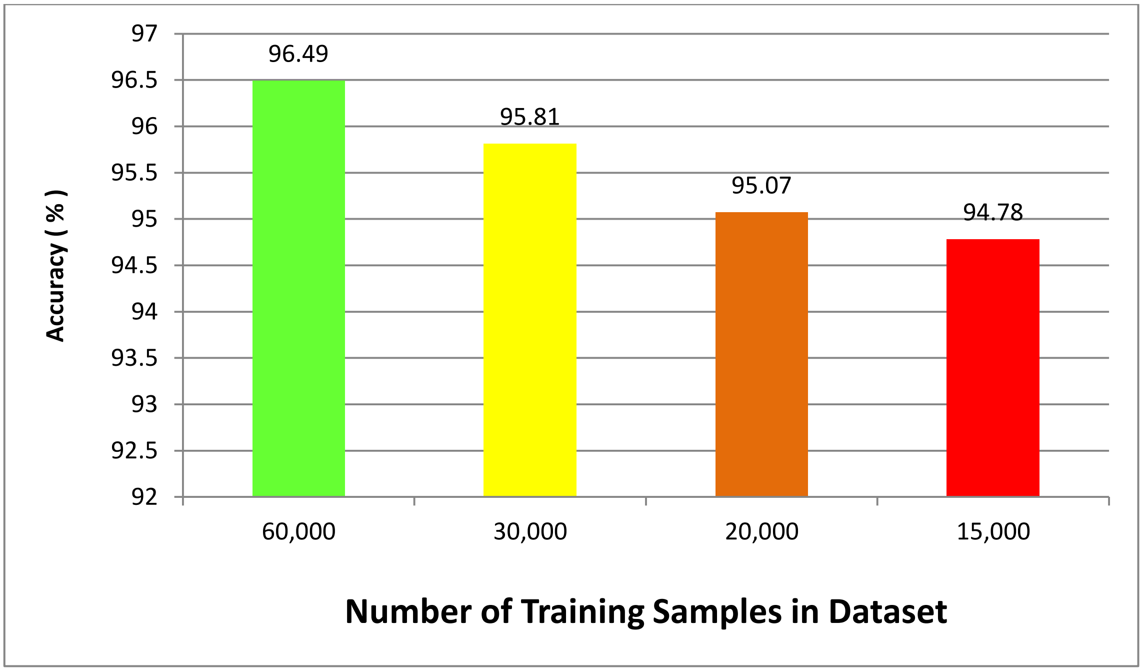


Figure 4.16 : Accuracy vs. number of training samples for recognition of Farsi Hoda dataset – digits part (k_NN classifier, k=1)

Table 4.2 : Recognition accuracy using different versions of training dataset – Hoda digits part

Number of Training Samples	Recognition Time for a Sample (Seconds)	Ratio of Recognition Time to initial Recognition Time T1	Accuracy (%) using *k*-NN classifier *k* = 1
60,000	T1 = 0.1161352	T1/T1 = 100%	96.49
30,000	T2 = 0.0509572	T2/T1 = 43.87748%	95.81
20,000	T3 = 0.0367815	T3/T1 = 31.67128%	95.07
15,000	T4 = 0.0296006	T4/T1 = 25.48805%	94.78

4.4.5.2 Experiments on Characters Part of the Hoda Dataset

By using sampling operation, three reduced training versions of Farsi Hoda dataset, characters part, are created: half (35,322 samples) by using sampling rate 1/2, one-third (23,548 samples) by using sampling rate 1/3, and one-fourth (17,661 samples) by using sampling rate 1/4. The original training dataset with 70,645 samples and these three reduced versions of datasets are used in the recognition stage. Here again, all the 400 pixels of an image were considered as input to a *k*-NN classifier. The final results are shown in Figure 4.17 and Table 4.3. As it is seen in Table 4.3, the recognition speed increases to about double, while the accuracy decreases from 80.67% to 79.36%.

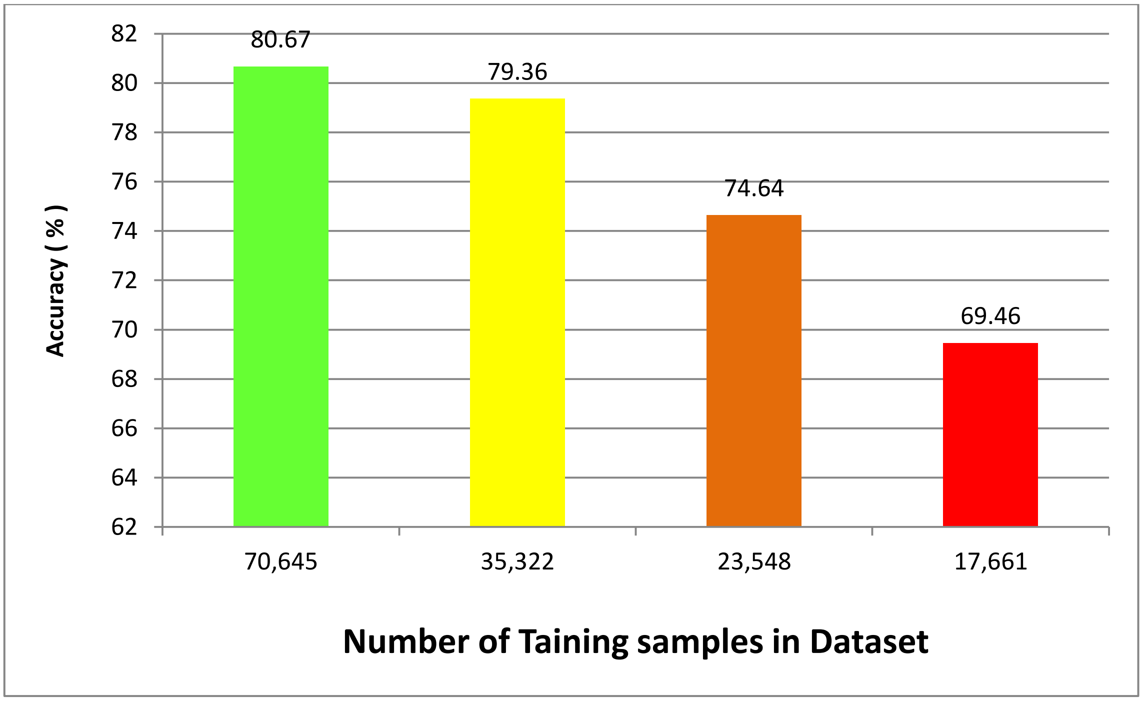


Figure 4.17 : Accuracy vs. number of training samples for recognition of Farsi Hoda dataset – characters part (k_NN classifier, k=1)

Table 4.3 : Recognition accuracy using different versions of training dataset – Hoda characters part

Number of Training Samples	Recognition Time for a Sample (Seconds)	Ratio of Recognition Time to initial Recognition Time T1	Accuracy (%) using *k*-NN classifier (*k*=1)
70,645	T1 = 0.0725580	T1/T1 = 100%	80.67
35,322	T2 = 0.0380366	T2/T1 = 52.4223%	79.36
23,548	T3 = 0.0282187	T3/T1 = 38.8912%	74.64
17,661	T4 = 0.0230077	T4/T1 = 31.7094%	69.46

4.4.5.3 Finding the Best Threshold for Sieving Operation

In order to investigate the relation between sieving operation rate and system accuracy, the aforementioned experiment in Section 4.4.5.1 was repeated again for Farsi digits. Here, the volume of training dataset was decreased from 100% to 20% with decreasing rate of 5%, by using the proposed SBR method. Figure 4.18 shows the relation between volume of reduced dataset and ratio of achieved accuracy to initial accuracy for this experiment. Also, Figure 4.19 shows the relation between volume of reduced dataset and value of recognition error for Farsi Hoda dataset, digits part. These figures indicate the best threshold for sieving Farsi Hoda dataset, digits part, is producing a 60% reduced version of initial dataset. In this case, the accuracy is decreased only 0.32%, from initial value 96.49% to new value 96.17% (the green arrow in the Figure 4.18). In this figure, the rate of accuracy loose was almost steady till the volume of reduced dataset was bigger than 60% of initil volume dataset. But, it was dropped significantly, when the size of reduced dataset became less than 60% of initial volume dataset. Hence, this point was considered as threshold for reducing Farsi digits volume dataset.

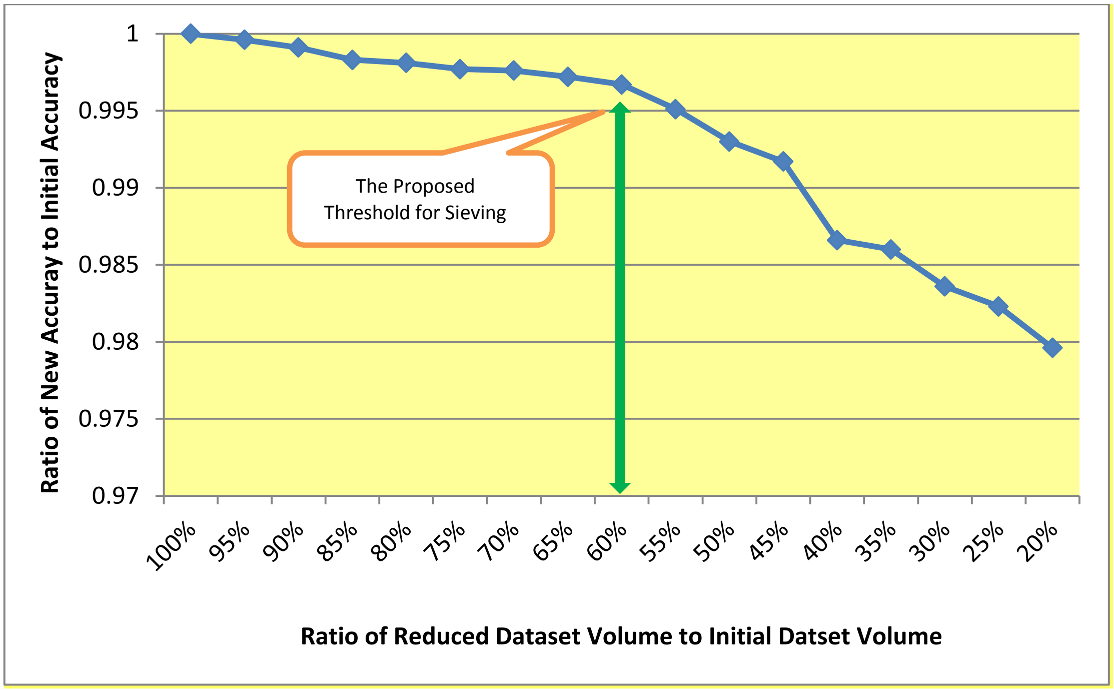


Figure 4.18 : Relation between dataset size reduction by SBR method and ratio of achieved accuracy to initial accuracy for Farsi Hoda dataset – digits part

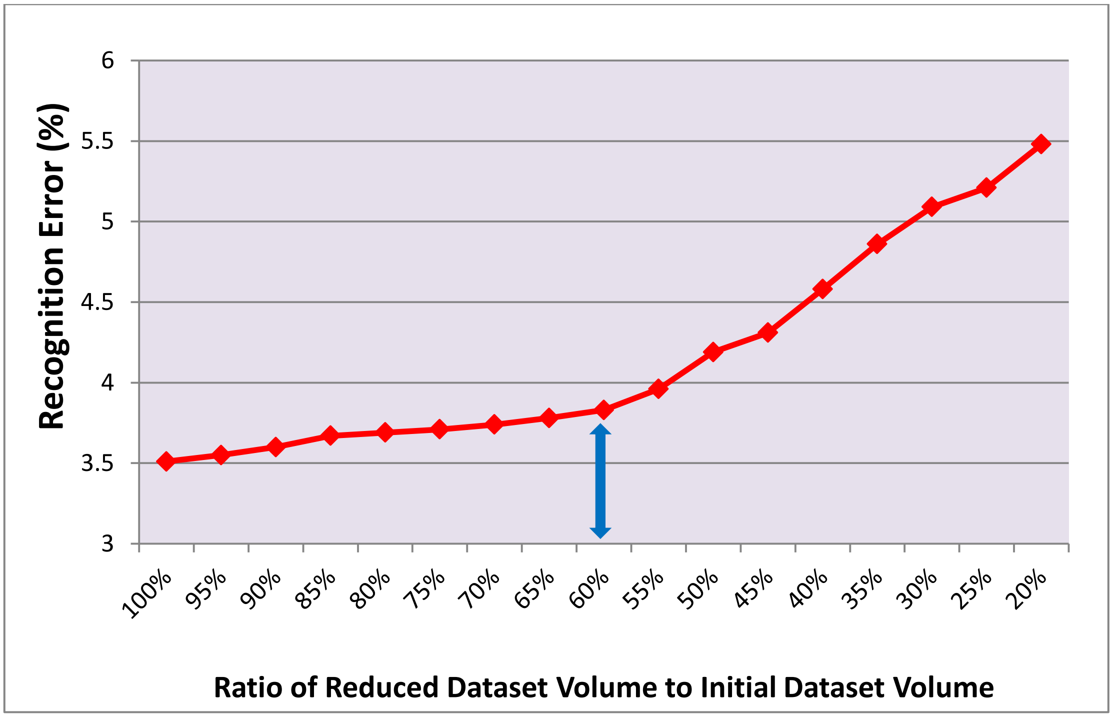


Figure 4.19 : Relation between dataset size reduction by SBR method and recognition error for Farsi Hoda dataset – digits part

The mentioned experiment was repeated for Farsi letters, corresponding to experiment in Section 4.4.5.2. Here, the volume of training dataset was decreased from 100% to 20% with decreasing rate of 5%, by using the proposed SBR method. Figure 4.20 shows the relation between the volume of reduced dataset and ratio of achieved accuracy to initial accuracy for this experiment. Also, Figure 4.21 shows the relation between volume of reduced dataset and value of recognition error for Farsi Hoda dataset, characters part. These figures indicate the best threshold for sieving Farsi Hoda dataset, characters part, is producing a 45% reduced version of initial dataset. In this case, the accuracy is decreased only 1.81%, from initial value 80.67% to new value 78.86%. Similar to Figure 4.18, the breaking point 45% - of the achieved accurecies - was considered as a threshold for Farsi letter size reduction.

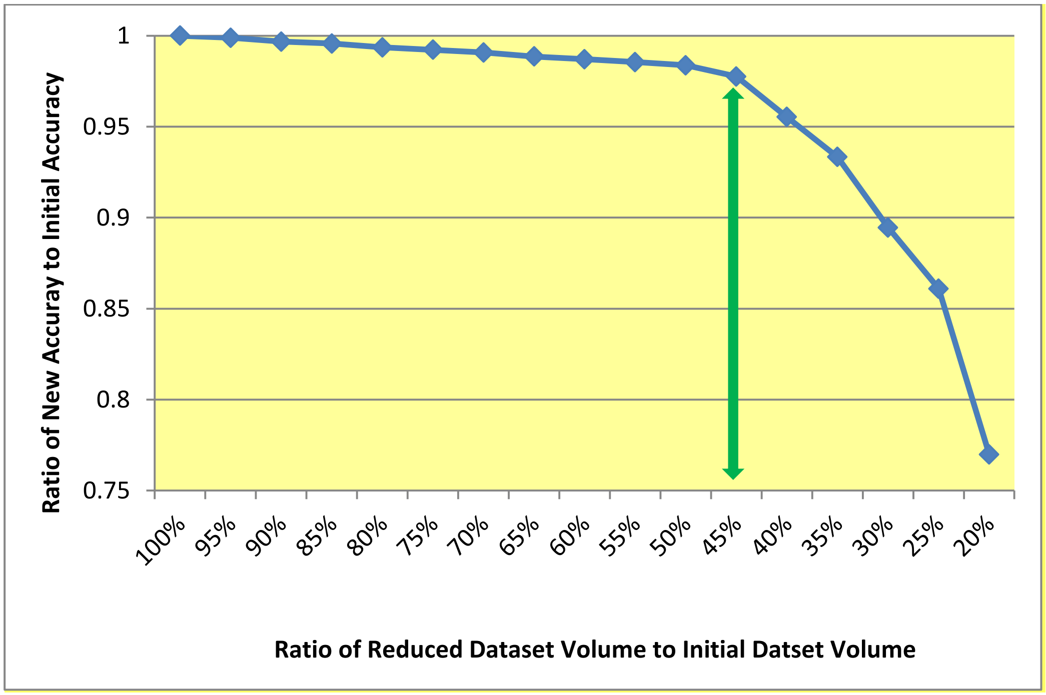


Figure 4.20 : Relation between dataset size reduction by SBR method and ratio of achieved accuracy to initial accuracy for Farsi Hoda dataset – characters part

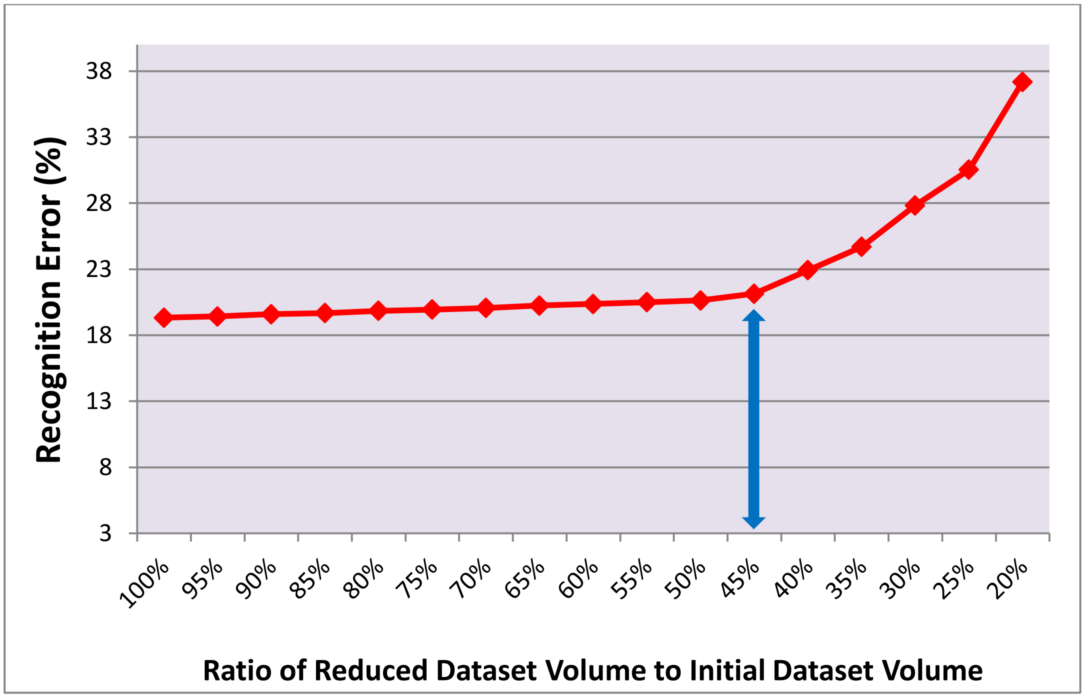


Figure 4.21 : Relation between dataset size reduction by SBR method and recognition error for Farsi Hoda dataset – characters part

4.5 Features Extraction

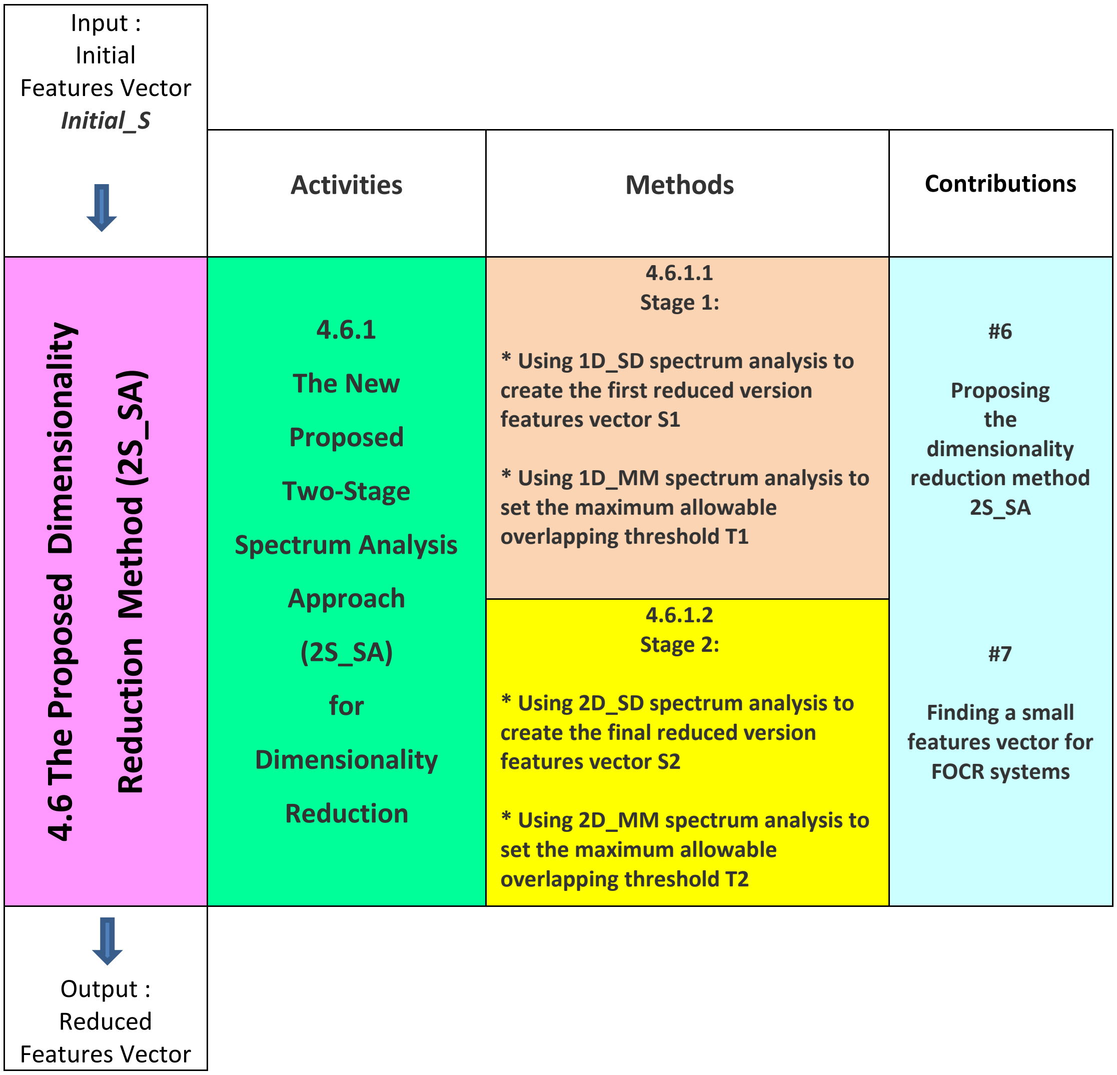
In PR issue, it is a common way to deal with some extracted features from the images instead of dealing with the original images, because the extracted features usually represent the image with smaller number of parameters.

In this thesis, via a comprehensive study on the literature in OCR area, an initial features vector, namely ***Initial_S,*** including 133 of different types of features (Table 1, Appendic VII) was extracted from the input samples and used in recognition step. Also, ***Initial_S*** is used as input for the proposed dimensionality reduction method 2S_SA (Section 4.6) to find a small set of features for handwritten FOCR systems. The second column of Table 1 in Appendix VII displays the all elements of the initial features vector ***Initial_S.***

4.6 Dimensionality Reduction

In this research, the dimensionality reduction operation is carried out to address the third objective: ***i.e. decreasing the system training and testing time and increasing the system accuracy.*** The contributions of this section are: ***1) Proposing the new two stages spectrum analysis (2S_SA) method for dimensionality reduction; 2) Finding a small effective features set for handwritten FOCR systems.***

Dimensionality reduction (feature selection) is an important step in PR systems. Although, there are various conventional approaches for feature selection, such as Principal Component Analysis, Random Projection, and Linear Discriminant Analysis, but selecting optimal, effective, and robust features is usually a difficult task (Section 2.5). In this thesis, a new two-stage method is proposed for dimensionality reduction purpose. This method is based on analysing one-dimensional and two-dimensional spectrum diagrams of standard deviation and minimum to maximum distributions for initial features vector elements. Figure 4.22 depicts the components of this module.



Input :
Initial
Features Vector
Initial_S

	Activities	Methods	Contributions
4.6 The Proposed Dimensionality Reduction Method (2S_SA)	**4.6.1 The New Proposed Two-Stage Spectrum Analysis Approach (2S_SA) for Dimensionality Reduction**	**4.6.1.1 Stage 1:** *** Using 1D_SD spectrum analysis to create the first reduced version features vector S1** *** Using 1D_MM spectrum analysis to set the maximum allowable overlapping threshold T1**	**#6** **Proposing the dimensionality reduction method 2S_SA**
		4.6.1.2 Stage 2: *** Using 2D_SD spectrum analysis to create the final reduced version features vector S2** *** Using 2D_MM spectrum analysis to set the maximum allowable overlapping threshold T2**	**#7** **Finding a small features vector for FOCR systems**

Output :
Reduced
Features Vector

Figure 4.22 : Overview of the proposed dimensionality reduction module 2S_SA

4.6.1 The New Proposed Two-Stage Spectrums Analysis (2S_SA) Method for Dimensionality Reduction

The constraint block 4.6 shows the proposed 2S_SA method for creating a smaller features vector ***S2***, from the initial features vector ***Initial_S***, symbolically.

$$\begin{cases} Initial\_S \leftarrow \ \ Features\ Extraction\ Opeartion\ (dataset) \\ S1 \leftarrow Stage\ 1\ of\ 2S\_SA(Initial\_S) \\ S2 \leftarrow Stage\ 2\ of\ \ 2S\_SA(S1) \end{cases} \qquad (4.6)$$

In constraint block 4.6, the following relation is established: ***S2* ≤ *S1* ≤ *Initial_S***. The proposed method 2S_SA utilizes One-Dimensional Standard Deviation (1D_SD) spectrum diagrams, Two-Dimensional Standard Deviation (2D_SD) spectrum diagrams, One-Dimensional Minimum to Maximum (1D_MM) spectrum diagrams, and Two-Dimensional Minimum to Maximum (2D_MM) spectrum diagrams analysis. To generate these spectrum diagrams, minimum, maximum, mean, and Standard Deviation (SD) values for all initial extracted features should be computed.

4.6.1.1 Stage 1 : Using One-Dimensional Spectrum Analysis Tool

a) One-Dimensional Standard Deviation (1D_SD) Spectrum

In 1D_SD spectrum, the mean and SD values, corresponding to a specific feature, by using all samples in different classes, are computed, firstly. Then, a spectrum line corresponding to that feature is drawn from mean-SD to mean+SD for each class. For example, there are 10 classes corresponding to digits ‘0’ to ‘9’ in digits recognition case. For creating 1D_SD spectrum diagrams, the mean, standard deviation (SD), mean-SD, and mean+SD are computed. Table 4.4 shows the mentioned values for ‘***Normalized Vertical Transition’*** feature of English digits (MNIST dataset). For more simplicity, the values were rounded to the nearest integer numbers.

Table 4.4 : Mean, SD, mean-SD, and mean+SD of '***Normalized Vertical Transition***' feature for English digits from MNIST dataset

Class (Digit)	**mean**	**Standard Deviation (SD)**	**mean-SD**	**mean+SD**
0	111	25	86	136
1	32	25	7	57
2	133	32	101	165
3	127	24	103	151
4	94	24	70	118
5	116	28	88	144
6	97	18	79	115
7	95	21	74	116
8	110	19	91	129
9	96	20	76	116

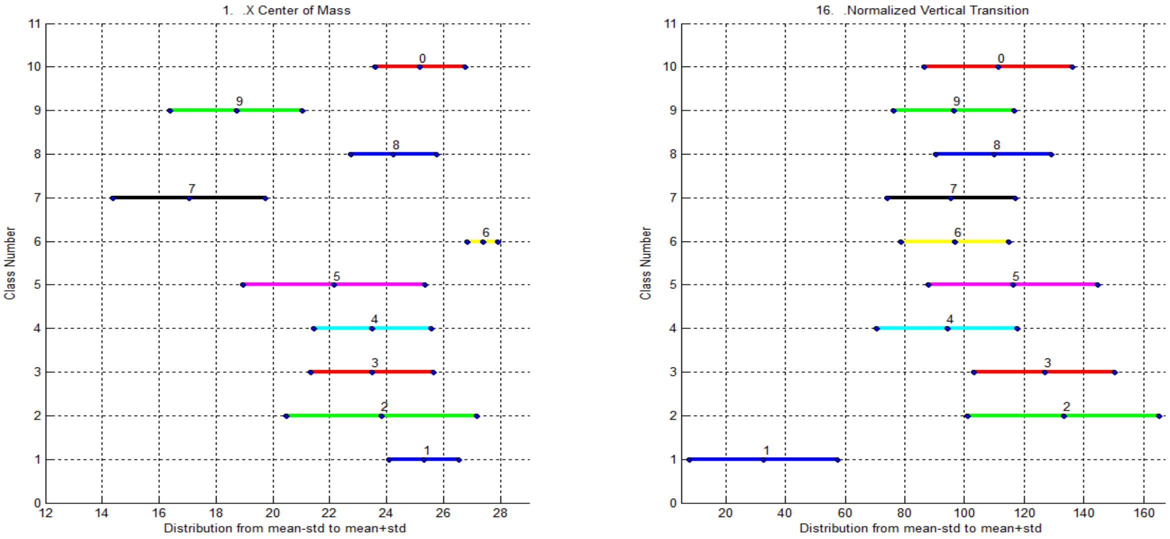
The length of 1D_SD spectrum line for a class is twice of its SD, corresponding to that feature. The smaller length for a spectrum line means the samples of that class are more similar to each other, respect to that specific feature, compared to a class with longer spectrum line. Two spectrum lines have overlapping, if the mean+SD of line '1' is greater than mean-SD of line '2' and mean+SD of line '2' is greater than mean-SD of line '1'.

Table 4.5 shows the convolution overlapping matrix of ***'Normalized Vertical Transition'*** feature for all pair classes of English digits (MNIST dataset). Each cell of this table indicates the value of spectrum lines overlapping for two classes respect to ***'Normalized Vertical Transition'*** feature. The smaller value of a cell is better than the larger value, because it means those classes have less overlapping. The value '0' indicates there is not any overlapping between those classes, and as a result, the related feature can separate those classes from each other, completely.

Table 4.5 : Convolution overlapping matrix of ***'Normalized Vertical Transition'*** feature for all pair classes of English digits (MNIST dataset)

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	-----	0	35	33	31	48	29	30	39	30
1	0	-----	0	0	0	0	0	0	0	0
2	35	0	-----	48	17	43	14	15	28	15
3	33	0	48	-----	15	41	12	13	26	13
4	31	0	17	15	-----	30	36	42	27	40
5	48	0	43	41	30	-----	27	28	38	28
6	29	0	14	12	36	27	-----	36	24	36
7	30	0	15	13	42	28	36	-----	25	40
8	39	0	28	26	27	38	24	25	-----	25
9	30	0	15	13	40	28	36	40	25	-----

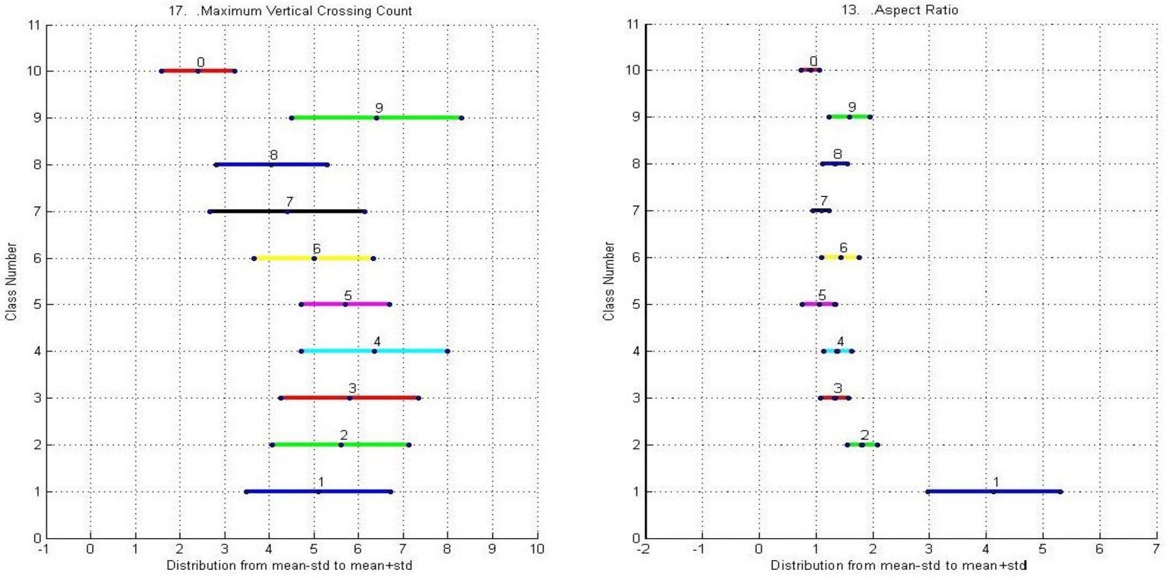
For digits recognition case, and in order to find the final reduced features vector, a 1D_SD diagram is plotted with 10 spectrum lines for each available feature in the initial features vector, corresponding to digits '0' to '9'. Figure 4.23.a and Figure 4.23.b show the 1D_SD distribution diagrams corresponding to '***X Coordinate Centre of Mass'*** and '***Normalized Vertical Transition'*** features for English digits, respectively. In Figure 4.23.a, the majority of spectrum lines are in an overlapping range [20 , 25], meaning that the '***X Coordinate Centre of Mass'*** feature, alone, cannot discriminate existing classes from each other in the features space. In Figure 4.23.b, the spectrum line corresponding to class (digit) '1' is completely separated from the other spectrum lines, indicating that the '***Normalized Vertical Transition'*** feature can completely discriminate digit (class) '1' from the other English digits (other classes). Therefore, this feature can be considered as a candidate feature in the final features vector.



(a) ‘X Coordinate Centre of Mass’ feature (b) ‘Normalized Vertical Transition’ feature

Figure 4.23 : 1D_SD spectrums diagram for the English digits set

Similar to Figures 4.23, Figures 4.24.a and 4.24.b show the 1D_SD distribution diagrams corresponding to ‘***Maximum Vertical Crossing Count’*** and ‘***Aspect Ratio’*** features for the Farsi digits, respectively. In Figure 4.24.a, the majority of the spectrum lines are in an overlapping range [3.5 , 7], meaning that the ‘***Maximum Vertical Crossing Count’*** feature, alone, cannot discriminate the existing classes from each other in the features space. In Figure 4.24.b, the spectrum line corresponding to class (digit) ‘1’ is completely separated from other spectrum lines, indicating that the ‘***Aspect Ratio’*** feature can completely discriminate class (digit) ‘1’ from the other Farsi digits (other classes). Therefore, it can be considered as a candidate feature in the final features vector.



(a) ‘Maximum Vertical Crossing Count’ feature (b) ‘Aspect Ratio’ feature

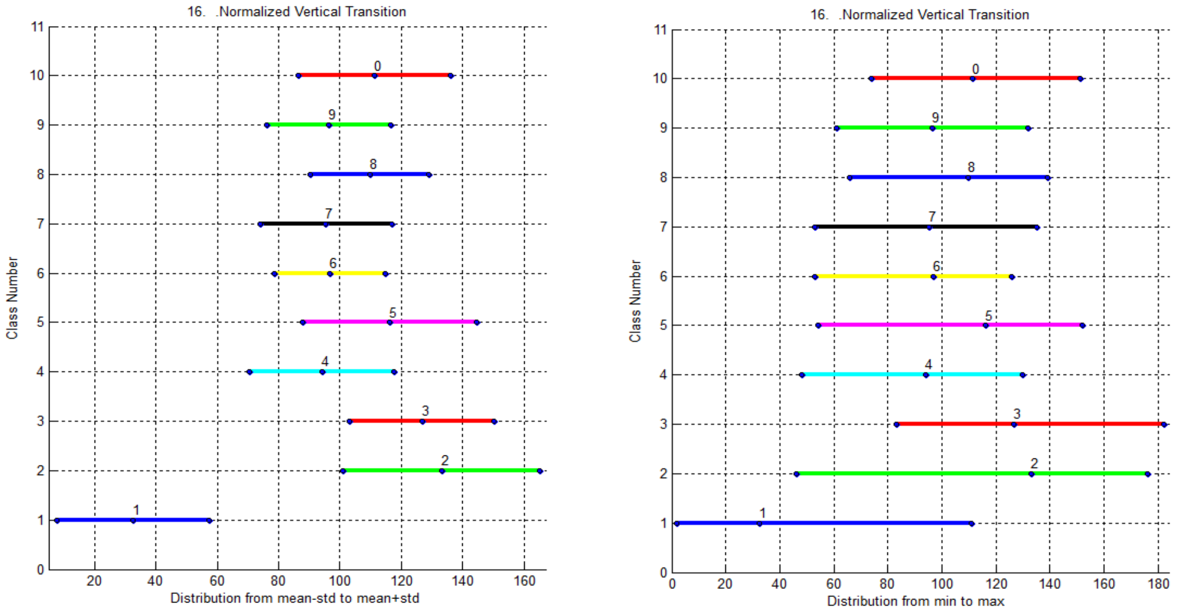
Figure 4.24 : 1D_SD spectrums diagram for the Farsi digits set

A shorter spectrum line corresponding to a specific feature indicates that the existing samples in a particular class have more similarity (less diversity) to each other in respect to that feature. In addition, a distribution diagram with class centers (locations of the means of the classes) farther apart is better than one with closer class centers. In this case, a classifier separates the existing clusters better.

b) One-Dimensional Minimum to Maximum (1D_MM) Spectrum

Finding a set of separated spectrum lines by using 1D_SD distribution diagrams is not enough to create an optimum features vector, because the outlier samples in each class are not in the range of 1D_SD spectrum lines. Indeed, they are in the 1D_MM range. In the 1D_MM plot, the minimum and maximum values corresponding to a specific feature, using all samples in different classes, are computed, firstly. Then, a spectrum line corresponding to that feature is drawn from the minimum to the maximum value of that specific feature for each class. Meanwhile, the other conditions are similar to 1D_SD plots.

Figures 4.25.b displays 1D_MM spectrums diagram for the feature '***Normalized Vertical Transition'*** (Figure 4.25.a) for English digits set. It is obvious that in this figure, some samples of class '1' overlap with some samples in all of the rest classes. This means that in the recognition phase, these samples may be misclassified to other classes and vice versa, if only the '***Normalized Vertical Transition'*** feature is employed.



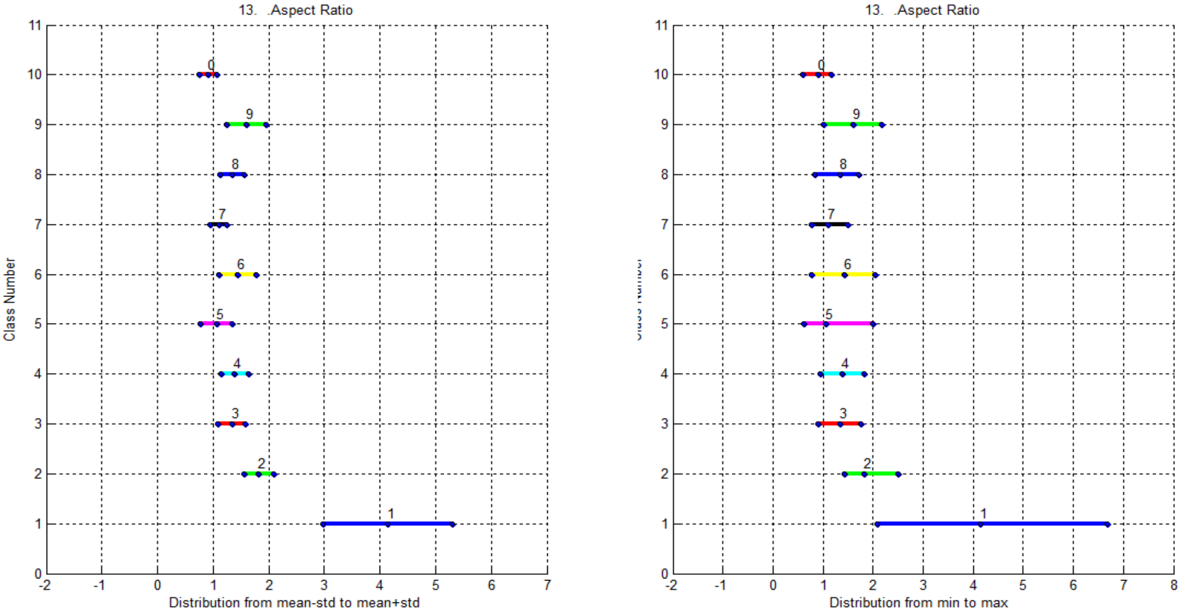
(a) (b)

(a) 1D_SD Spectrums diagram for 'Normalized Vertical Transition' feature corresponding to English digits

(b) 1D_MM Spectrums diagram for 'Normalized Vertical Transition' feature corresponding to English digits

Figure 4.25 : Comparing 1D_SD and 1D_MM spectrum distribution diagrams

Figures 4.26.b displays 1D_MM spectrum lines for the feature '***Aspect Ratio'*** (Figure 4.26.a) for Farsi digits set. It is obvious that in this figure, some samples of class '1' overlap with some samples in classes '2' or '9'. In other words, in the recognition phase, it is possible that some samples of class '1' are misclassified to classes '2' or '9' and vice versa, if only the '***Aspect Ratio'*** feature is utilized.



(a) (b)

(a) 1D_SD Spectrums diagram for ‘Aspect Ratio’ feature corresponding to Farsi digits

(b) 1D_MM Spectrums diagram for ‘Aspect ratio’ feature corresponding to Farsi digits

Figure 4.26 : Comparing 1D_SD and 1D_MM spectrum distribution diagrams

In the proposed dimensionality reduction method 2S_SA, 1D_MM is used to find the maximum allowable overlapping threshold ***T_1***, to create the first reduced features vector ***S1*** from the initial features set ***Initial_S***. By investigating the overlapping values of the spectrum lines in the 1D_MM diagram for each feature in ***Initial_S***, the value of threshold ***T_1*** is selected. In this study, the ***T_1*** threshold was selected 30%, experimentally.

4.6.1.2 Stage 2 : Using Two-Dimensional Spectrum Analysis Tool

a) Two-Dimensional Standard Deviation (2D_SD) Spectrum

Similar to the 1D_SD and 1D_MM distribution diagrams, the Two-Dimensional Standard Deviation (2D_SD) spectrums diagram and the Two-Dimensional Minimum to Maximum (2D_MM) spectrums diagram for two features are made by mapping one feature on the *X* axis and another feature on the *Y* axis. In these cases, an ellipse (or rectangular) is plotted for each couple of features. In 2D_SD, the main ellipse diagonals (or the length and the width in the rectangular case) are plotted from the mean-SD to the mean+SD for two features. In 2D_MM, the main ellipse diagonals (or the length and the width in the rectangular case) are plotted from the minimum value to the maximum value for those two features. As such, the [n×(n-1)/2] 2D_SD (or 2D_MM) distribution diagram can be generated for *n* independent features.

Figure 4.27 shows a 2D_SD distribution diagram for two features; namely, '***X Coordinate Centre of Mass***' and '***No. of Foreground Pixels in Upper Half of Image'*** for the Farsi digits set. It is completely clear that the ellipse for class (digit) '0' is completely distinct from the other ellipses. Hence, the feature pair ('***X Coordinate Centre of Mass'***, **'*No. of Foreground Pixels in Upper Half of Image'***) is a good choice for membership in the final features vector (to distinguish class (digit) '0' from the other classes (digits)).

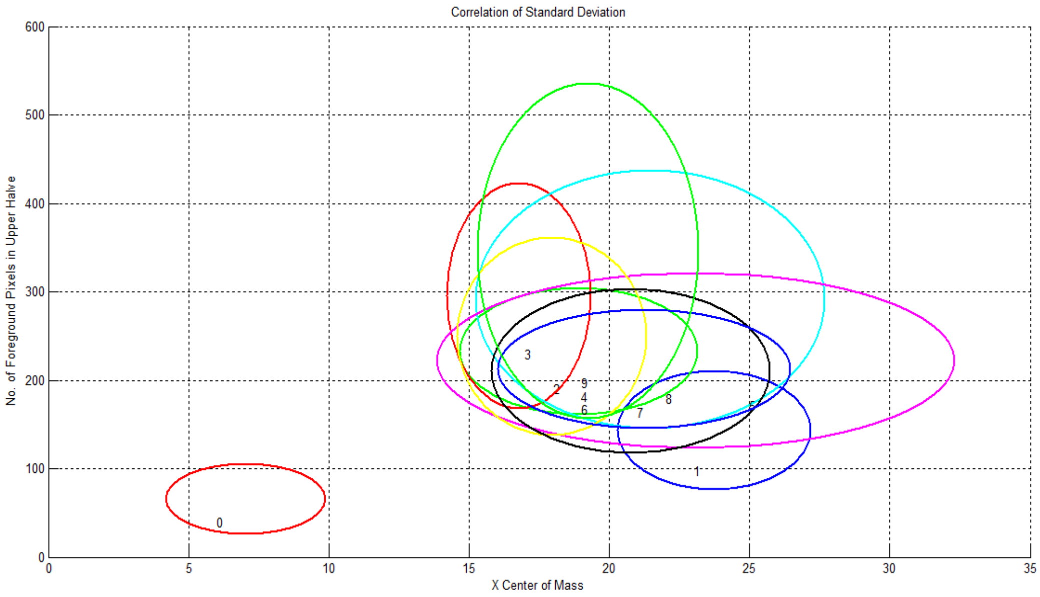


Figure 4.27 : 2D_SD spectrums distribution diagram for Farsi digits
'X Coordinate Center of Mass' feature vs.
'No. of Foreground Pixels in Upper Half of image' feature

Figure 4.28 shows another example for 2D_SD distribution of two features: ‘***Y Coordinate Centre of Mass***’ and ‘***No. of Foreground Pixels in Lower Half of Image’*** of the Farsi digits set. It is completely clear that, in this case, the mentioned features are highly correlated. Hence, they are not a suitable feature pair for membership in the final features vector.

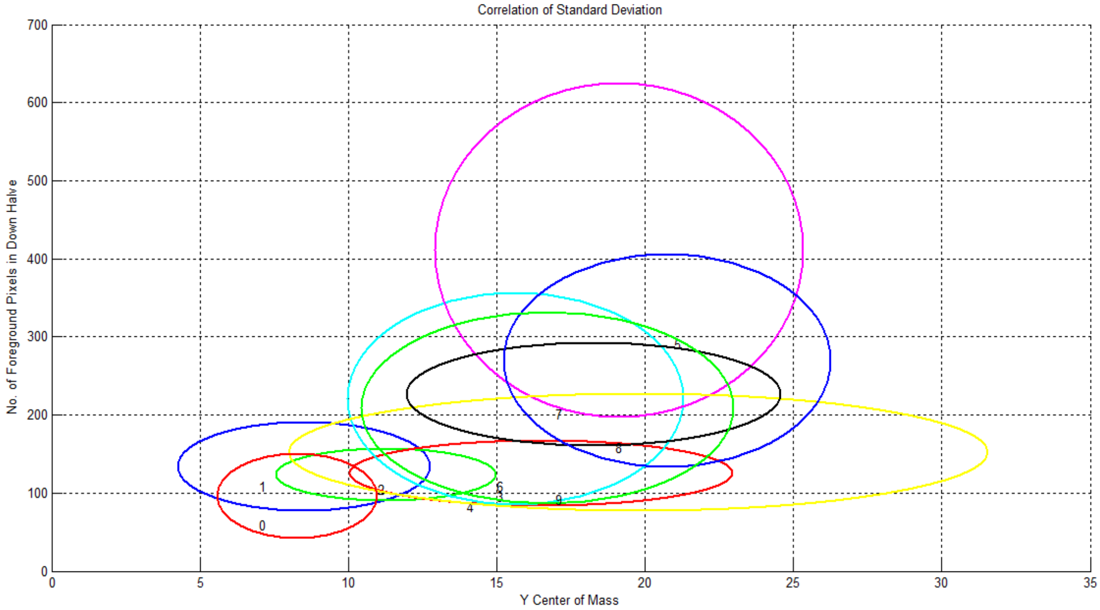


Figure 4.28 : 2D_SD spectrums distribution diagram for Farsi digits
‘Y Coordinate Centre of Mass’ feature vs.
‘No. of Foreground Pixels in Lower Half of image’ feature

b) Two-Dimensional Minimum to Maximum (2D_MM) Spectrum

In the proposed dimensionality reduction method, 2D_MM is utilized to find the maximum allowable overlapping threshold $\boldsymbol{T_2}$, to create the final reduced features vector ***S2*** from the first reduced features vector ***S1***. By investigating the overlapping values of the spectrum ellipses (rectangular) in the 2D_MM diagram for all features pair in ***S1***, the value of threshold $\boldsymbol{T_2}$ is selected. In this study, the $\boldsymbol{T_2}$ threshold was selected 20%, experimentally.

By using the literature and based on the introduced features in Section 4.5, an initial features vector ***Initial_S*** with 133 members of most-used features, which are employed for handwritten characters recognition, was identified and extracted from the training samples (The second column of Table 1 of Appendix VII). For the dimensionality reduction process, the value of a specific feature is defined as $f_k(S_{i,j})$, where f_k is the value of the k'th feature from the initial features vector ***Initial_S,*** and $S_{i,j}$ represents the j'th sample of class i. Subsequently, and by using all samples in the training part of each class, the values of the minimum, maximum, mean, and standard deviation for all features in the initial features vector are computed. The reduction operation is carried in two stages as follows:

Stage 1: Using 1D_SD and 1D_MM

To find the first reduced subset of features vector, the 1D_SD and 1D_MM distribution spectrums are generated for all available features in initial features vector ***Initial_S***. The system selects every feature that its 1D_SD spectrum line had a maximum of 30% overlapping (threshold $\boldsymbol{T_1}$, that it was found experimentally) with the other 1D_SD spectrum lines of the other classes. The output of this stage is the first reduced version of the features vector ***S1***, which satisfies the criteria necessary for membership in the final features vector (the columns 4, 6, and 8 of Table 1 in Appendix VII).

Stage 2: Using 2D_SD and 2D_MM

By plotting the 2D_SD and 2D_MM distribution diagrams for the first reduced version of the features vector ***S1***, the final reduced version of features vectors ***S2*** is created (the columns 5, 7, and 9 of Table 1 in Appendix VII). In this stage, a couple of features are selected, if their 2D_SD has a maximum of 20% overlapping (threshold $\boldsymbol{T_2}$, that it was found experimentally) with the other 2D_SD distribution diagrams of the other classes. In this stage, it is possible that a feature is added to ***S2*** more than once. Hence, in the end, the repetitive features in ***S2*** are removed to create the smallest size of ***S2***.

In the proposed dimensionality reduction method 2S_SA, the 1D_SD and 2D_SD spectrums are utilized to decide whether or not a feature is suitable for membership in the final features vector. 1D_MM and 2D_MM are used to find the best value for thresholds T_1 and T_2. It is obvious that these threshold values are dependent to the type and characteristics of training dataset samples. Algorithm 3 explains the first stage of applying 2S_SA method to generate the intermediate reduced features vector ***S1*** from initial features vector ***Initial_S.*** The output of this stage is fed as input to next stage (Algorithm 4).

Algorithm 3. 2S_SA method for dimensionality reduction purpose – Stage 1.

Input:

```
        Initial_S : Initial Features Vector  /*  Initial_S = {f_1, f_2, … f_n}  */
        n : Number of features in the initial features vector , i.e. Initial_S
        noc : Number of Classes in Pattern Space
```

Output:

```
        S1 : First Reduced Version of Features Vector  /*  S1 = {g_1, g_2, … g_k}  ;  k ≤ n  */
```

Method: 2S_SA

```
S1:= null ;
for  k = 1 : n do
{
      Compute the coordinate of all 1D_SD spectrum lines corresponding to
                  feature f_k ;
      for  c = 1 : noc do
      {
          If ( overlapping of spectrum line of class c with all the rest
                  spectrum lines has the value less than threshold T1 ) then
            {
                  Insert feature f_k to S1;
                  goto L1;
            }
       }
  L1: continue;
  }
  return (S1);
```

Algorithm 4 explains the second stage of applying 2S_SA method to generate the final reduced features vector ***S2,*** from the first reduced features vector ***S1***. The input for this step is the output of stage 1, and the output of this stage is the final reduced features vector.

Algorithm 4. 2S_SA method for dimensionality reduction purpose – Stage 2

Input:

S1 : First Reduced version of Features Vector /* ***S1*** = $\{g_1, g_2, \dots g_k\}$ */
k : Number of features in the first reduced features vector, i.e. ***S1***
noc : Number of Classes in Pattern Space

Output:

S2* :** Final Reduced Features Vector /* ***S2 = $\{e_1, e_2, \dots e_j\}$; $j \le k$ */

Method: 2S_SA

```
m := Number of features in the first reduced version of the features vector S1;
S2:= null ;
for k = 1 : m do
      for h = 1 : m do
          {
              Compute the coordinate of all 2D_SD spectrum ellipses
                          corresponding to features pair ($f_k$ , $f_h$);
              for c = 1 : number of classes do
              {
                  if ( overlapping of spectrum ellipses of class c with all the rest
                     spectrum ellipses has the value less than threshold T2 ) then
                  {
                        Insert feature pair $f_k$ and $f_h$ to S2;
                        goto L2;
                  }
              }
          }
      L2: continue;
}
delete the repetitive features from S2;
return (S2) ;
```

The mentioned operations created the features vectors ***Initial_S***, ***F/D-S1***, and ***F/D-S2*** for the digits part of the Farsi Hoda dataset, features vectors ***Initial_S***, ***F/C-S1***, and ***F/C-S2*** for the characters part of the Farsi Hoda dataset, and the features vectors ***Initial_S***, ***E-S1***, and ***E-S2*** for the English digits MNIST dataset. Table 4.6 shows the number of features in different versions of features vectors in each stage.

Table 4.6 : Number of features in initial features vector, first reduced version of features vector, and final reduced version of features vector, created by 2S_SA method

	Number of Features			
Dataset	Initial features vector : ***Initial_S***	First reduced versions of features vector : ***S1***	Final reduced versions of features vector : ***S2***	Ratio of ***S2*** to ***Initial_S***
Farsi dataset Hoda Digits part	133	***F/D-S1*** : 94	***F/D-S2*** : 58	**0.44**
Farsi dataset Hoda Characters part	133	***F/C-S1*** : 119	***F/C-S2*** : 93	**0.70**
English dataset MNIST	133	***E-S1*** : 103	***E-S2*** : 79	**0.59**

4.6.2 Experimental Results of using 2S_SA Method

To investigate the efficiency of the proposed dimensionality reduction technique 2S_SA and finding the effect of this method on recognition accuracy, several experiments were carried out on Hoda and MNIST datasets. The following sub-sections demonstrate the experiments. Just for classifiers comparison in recognition operation, experiments were carried out with using three classifiers *k*-NN, NN, and SVM. However, the goal of this study was not classifiers comparison. Hence, the differences between the obtained results were not highlighted.

In the case of using *k*-NN classifier, the value of k is set to 1. According to Alaei, Nagabhushan and pal (2009), SVM classifier with Gaussian kernel is selected. In the case of using NN classifiers, each experiment is repeated 10 times, and finally the accuracy is reported in average. However, the details of experiments with using NNs are demonstrated in this section, and for two other classifiers, only the final results are reported. Also, in this step, some extra experiments are carried out by using k-fold cross validation (CV) technique (Section 3.4). According to Kohavi (1995), the value of k is set to 10. To such an aim, all the samples in training and testing parts are combined, then the whole dataset are divided to 10 folds, 9 folds are used for system training and the remainder fold is used for system testing. Finally, the average of 10 obtained recognition accuracies is reported as final accuracy. It should be mentioned that, in this section, the results are reported regardless of dataset size reduction operation, to show only the effect of dimensionality reduction operation on accuracy.

4.6.2.1 Experiments on Farsi Digits

In the first part, the proposed method 2S_SA was used in a FOCR system which it was employed to recognize digits part of the Hoda dataset. In all experiments, a MLP-NN was trained with 133, 94, or 58 (Table 4.6) neurons in the input layer (corresponding to the number of features in vectors ***Initial_S*** (or ***F/D-S1*** or ***F/D-S2)***, 30 neurons (found experimentally) in the hidden layer, and 10 neurons (corresponding to 10 different classes of digits ‘0’ to ‘9’) in the output layer, respectively. Also, in all experiments, the network was trained with all 60,000 samples from the training part, and was then tested with all 20,000 samples from the testing part of the Hoda dataset.

In the first experiment, the features vector ***Initial_S*** with all 133 features was used for training and testing the system. Finally, the system achieved to 90.41% accuracy. In the second experiment, the system was trained with the reduced proposed version of features vector ***F/D-S1*** with only 94 features. On average, the correct recognition rate increased from 90.41% to 92.60%, clearly indicating the superiority of the reduced features vector ***F/D-S1*** of the proposed method against the initial features vector ***Initial_S*** with 133 features. In the last experiment, the system was trained with the final reduced proposed version of features vector ***F/D-S2*** with only 58 features. The accuracy again increased from 92.60% to 95.12%, clearly indicating the superiority of the final reduced features vector ***F/D-S2*** of the proposed method against the initial features vector ***Initial_S*** with 133 features, and first reduced features vector ***F/D-S1*** with 94 features. The corresponding results are shown in Figure 4.29.

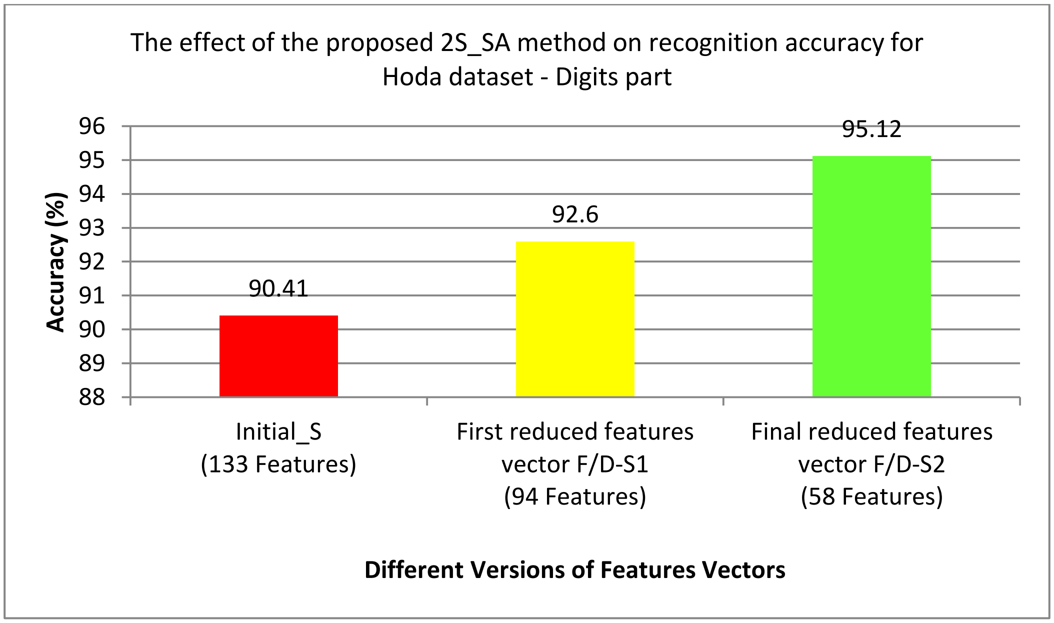


Figure 4.29 : Recognition rate corresponding to different versions of features vectors for digits part of Farsi Hoda dataset (ANN classifier)

4.6.2.2 Experiments on Farsi Letters

In the second part, all the experiments in section 4.6.2.1 were repeated for the characters part of Farsi Hoda dataset. ***Initial_S, F/C-S1***, and ***F/C-S2*** were the initial features vector with 133 features, first reduced features vector with 119 members, and the final reduced features vector including 93 features, in order (Table 4.6). In all experiments, a MLP-NN was trained with 133, 119, or 93 neurons in the input layer (corresponding to the number of features in vectors ***Initial_S, F/C-S1,*** or ***F/C-S2***, 50 neurons (found experimentally) in the hidden layer, and 36 neurons (corresponding to 36 different classes of available Farsi letters in alone mode) in the output layer, respectively. Also, in all experiments, the network was trained with all 70,645 samples from the training part, and was then tested with all 17,706 samples from the testing part of the Hoda dataset.

In this case, by using the proposed feature selection method 2S_SA, the number of features was reduced from 133 to 93; meanwhile the final accuracy increased from 81.82% to 83.74%, clearly indicating the superiority of the final reduced features vector ***F/C-S2*** against the initial features vector ***Initial_S*** with 133 features, and first reduced features set ***F/C-S1*** with 119 features. The corresponding results are shown in Figure 4.30.

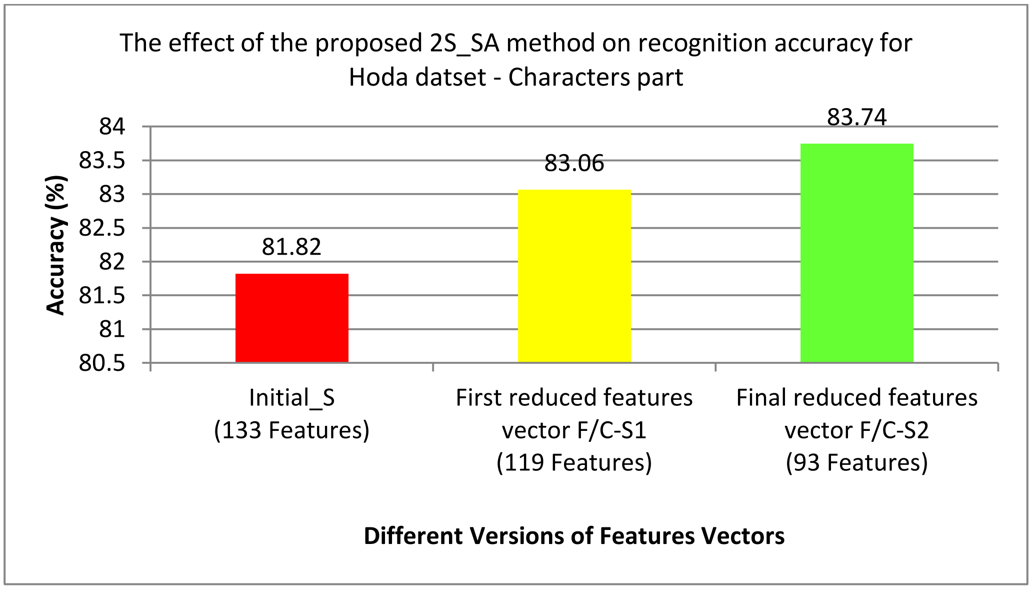


Figure 4.30 : Recognition rate corresponding to different versions of features vectors for characters part of Farsi Hoda dataset (ANN classifier)

4.6.2.3 Experiments on English Digits

In the last part, all the experiments in section 4.6.2.1 were repeated for the MNIST dataset to recognize handwritten English digits. ***Initial_S, E-S1***, and ***E-S2*** were the initial features vector with 133 features, first reduced features vectors with 103 members, and the final reduced features vector including 79 features, in order (Table 4.6). In all experiments, a MLP-NN was trained with 133, 103, or 79 neurons in the input layer (corresponding to the number of features in vectors ***Initial_S, E-S1,*** and ***E-S2)***, 30 neurons (found experimentally) in the hidden layer, and 10 neurons (corresponding to 10 different classes of digits '0' to '9') in the output layer, respectively. Also, in all experiments, the network was trained with all 60,000 samples from the training part, and was then tested with all 10,000 samples from the testing part of the MNIST dataset.

In this case, by using the proposed feature selection method 2S_SA, the number of features was reduced from 133 to 79; meanwhile the final accuracy increased from 91.93% to 94.88%, clearly indicating the superiority of the final reduced features vector ***E-S2*** of the proposed method against the initial features vector ***Initial_S*** with 133 features, and initial reduced features vector ***E-S1*** with 103 features. The corresponding results are shown in the Figure 4.31.

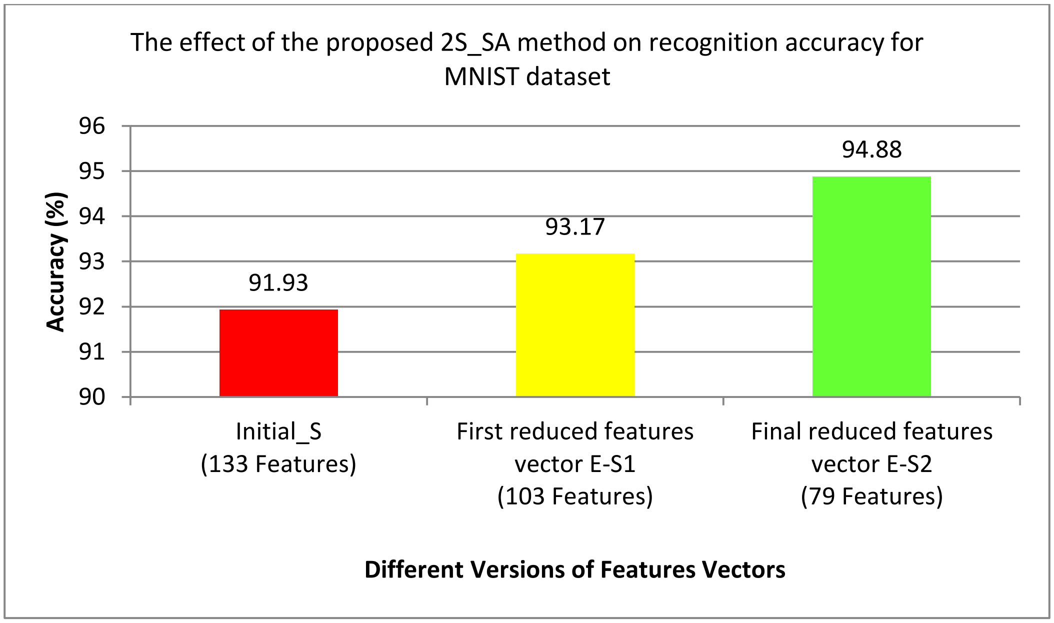


Figure 4.31 : Recognition rate corresponding to different versions of features vectors for English MNIST dataset (ANN classifier)

The obtained results of applying 2S_SA method, for dimensionality reduction purpose, on datasets Hoda and MNIST are plotted in Figure 4.32.

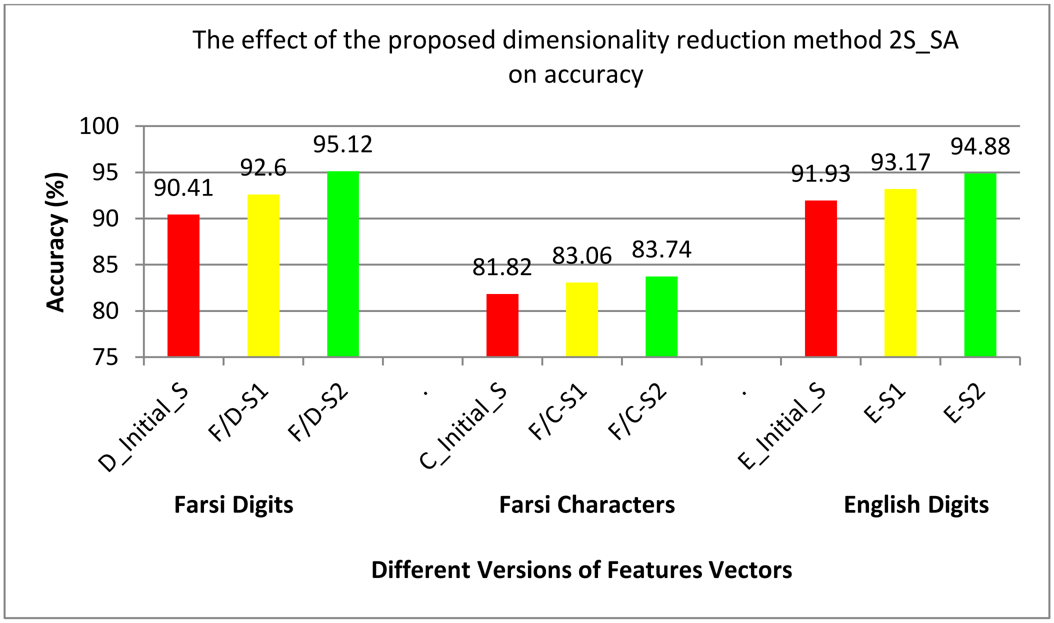


Figure 4.32 : Recognition rate corresponding to different versions of features vectors for Farsi dataset Hoda and English dataset MNIST (ANN classifier)

Table 4.7 shows the achieved accuracies, in this part, not only by employing NN as classifier, but also by using *k*-NN and SVM classifiers, too. The 2S_SA method succeeded to reduce the features vectors dimension; meanwhile the systems accuracies were increased.

Table 4.7 : The effect of applying the proposed dimensionality reduction method 2S_SA on recognition accuracy

Dataset	Classifier	# of Features in Initial features vector ***Initial_S***	Acc. %	# of Features in First reduced features vector ***S1***	Acc. %	# of Features in Final reduced features vector ***S2***	Acc. %
Farsi dataset Hoda Digits part	MLP-NN	133	90.41	94	92.6	58	95.12
	k-NN	133	91.13	94	93.02	58	95.88
	SVM	133	97.67	94	98.51	58	98.95
Farsi dataset Hoda Characters part	MLP-NN	133	81.82	119	83.06	93	83.74
	k-NN	133	79.07	119	80.26	93	81.12
	SVM	133	85.40	119	87.19	93	88.26
English dataset MNIST	MLP-NN	133	91.93	103	93.17	79	94.88
	k-NN	133	92.41	103	94.59	79	95.61
	SVM	133	98.40	103	98.69	79	98.78

In order to investigate the effect of using k-fold CV technique on recognition accuracy, the last part of experiments, i.e. using reduced features vectors ***F/D-S2***, ***F/C-S2***, and ***E-S2***, were repeated again in 10-fold CV (Kohavi, 1995). For digits part of the Hoda dataset, all the available samples in training part (60,000 samples) and testing part (20,000 samples) were combined to make a dataset with 80,000 samples. Then, it divided to 10 folds, each fold including 8,000 samples. For Characters part of the Hoda dataset, all the available samples in training part (70,645 samples) and testing part (17,700 samples) were combined to make a dataset with 88,000 samples (for more simplicity). Then, it divided to 10 folds, each fold including 8,800 samples. For MNIST dataset, all the available samples in training part (60,000 samples) and testing part (10,000 samples) were combined to make a dataset with 70,000 samples. Then, it divided to 10 folds, each fold including 7,000 samples. Then, nine folds were used for training and one fold was used for testing. Each experiment was repeated 10 times with different folds in training and testing parts. The achieved accuracies in 10 experiments were averaged to find the final accuracy. Here, only NN classifier was used. Table 4.8 shows the obtained results in this part. Compared to reported results in Table 4.7, the results in Table 4.8 are better in all cases. The main reasons for this better results in this case are: 1) Using more samples for training the systems; 2) Using fewer samples for testing; and 3) The effect of averaging process to nullify the effect of weak results of some experiments, in final accuracy.

Table 4.8 : The effect of applying Cross Validation technique on recognition accuracy (ANN classifier)

Dataset	Number of features in final reduced features vector ***S2***	Accuracy without CV	Accuracy by using 10-fold CV	Accuracy Enhancement
Farsi dataset Hoda Digits part	**58** ***(F/D-S2)***	95.12%	96.92%	1.80%
Farsi dataset Hoda Characters part	**93** ***(F/C-S2)***	83.74%	86.51%	2.77%
English dataset MNIST	**79** ***(E-S2)***	94.88%	95.80%	0.92%

However, for the other experiments in this study, the CV technique was not applied, because: 1) In order to perform accurate comparison with the literature, it is necessary to employ the same training and testing samples. However, using CV technique will change the combination of training and testing samples in datasets. In this situation, the results comparison is not carried out in similar conditions. 2) Hoda and MNIST datasets are big datasets, and in this case, it is not usually a common approach to apply CV techniques on big datasets with a large number of training and testing samples.

4.7 Classification (Recognition)

In order to create the similar conditions with the literature and to carry out the fair comparison with the other available researches in FOCR domain, the famous classifiers *k*-NN, MLP-NN, and SVM are employed in the recognition stage. However, corresponding to any experiment in the literature, the similar classifier is utilized. Brief descriptions of these recognition engines have been introduced in Section 2.3.4.

4.8 Summary

In this chapter, the technique CBP was introduced in order to connect broken parts of Farsi and English digits images (Section 4.3.3). This pre-processing operation increased the recognition accuracy 6.63% and 5.85% for Farsi and English digit recognition (Table 4.1).

To decrease the size of a training dataset, in order to increase the system speed, SBR technique was proposed (Section 4.4). First, a template was created for each class using modified frequency diagram concept (Section 4.4.1). Then, templates were converted to binary format (Section 4.4.2). Thereafter, by using similarity measurement function 4-5, a similarity value was computed for a sample. The existing samples in a dataset were sorted based on their similarity values. Finally, by using sieving operation, only some samples of each class were moved to final reduced dataset, with a specific sampling rate (Section 4.4.4). The SBR method decreased the size of training dataset to half, for both Farsi digits and letters datasets, and increased the recognition speed significantly, without any noticeable decreasing in system accuracy (Figure 4.16, Figure 4.17, Table 4.2, and Table 4.3).

In order to decrease the dimensionality of features space, the dimensionality reduction operation took place via the proposed method 2S_SA (Section 4.6.1). To such an aim, 1D_SD, 1D_MM, 2D_SD, and 2D_MM spectrum diagrams analysis were proposed and employed. The proposed 2S_SA method succeeded to decrease the number of features in features space, considerably (the last column of Table 4.6). The proposed methods were applied on two big benchmark datasets: digits and letters part of the handwritten Farsi Hoda dataset, and handwritten digits MNIST dataset, and the obtained results were reported. The 2S_SA method not only decreased the overall processing time, but also it increased the recognition accuracy for all experiments (Figure 4.29, Figure 4.30, and Figure 4.31).

CHAPTER 5

RESULTS COMPARISON AND DISCUSSION

5.1 Introduction

In this chapter, in order to evaluate the effectiveness of the proposed dataset size reduction method SBR (Section 4.4) and the proposed dimensionality reduction method 2S_SA (Section 4.6), the obtained results by these methods are compared to the literatures.

5.2 Results Comparison for Dataset Size Reduction Operation

The majority of available researches in dataset size reduction issue do not include the accurate information regarding the type of data and number of training samples after reduction operation. Most of those researches only reported some numerical results, based on the elapsed training time, before and after dataset reduction process, by proposing some new versions of SVM classification engine (Cano, Garcia & Herrera, 2008; Javed, Ayyaz & Mehmoud, 2007; Zhongdong, Jianping, Weixin & Xinbo, 2004). Only the research by Vishwanathan and Murty (2004), and the research by Cervantes, Li and Yu (2008) include the details of output results. However, the type of data in the later research is not the image for OCR application. Nonetheless, the mentioned researches are compared to other researches in OCR domain in Section 5.2.1.

5.2.1 Digits Samples

5.2.1.1 Vishwanathan Approach

Vishwanathan and Murty (2004) used support vector concept to categorize training samples of handwritten English digit ‘OCR’ dataset, including 6,670 samples. After finding the separator planes between the classes, they deleted boundary patterns, which were most likely to cause confusion during the classification operation, and also samples of a class which were far from the classes' boundaries. They succeeded in reducing the initial training samples size from 100% to 46.69%, 43.09%, 32.29%, and 25.35%, with decreasing system accuracy from 92.50% to 89.56%, 88.90%, 86.86%, and 84.88%, respectively (the first rows block of Table 5.2). Figure 5.1 shows the accuracy decreasing vs. dataset size reduction, and also, Figure 5.2 shows the accuracy decreasing ratio vs. dataset size reduction ratio for this experiment.

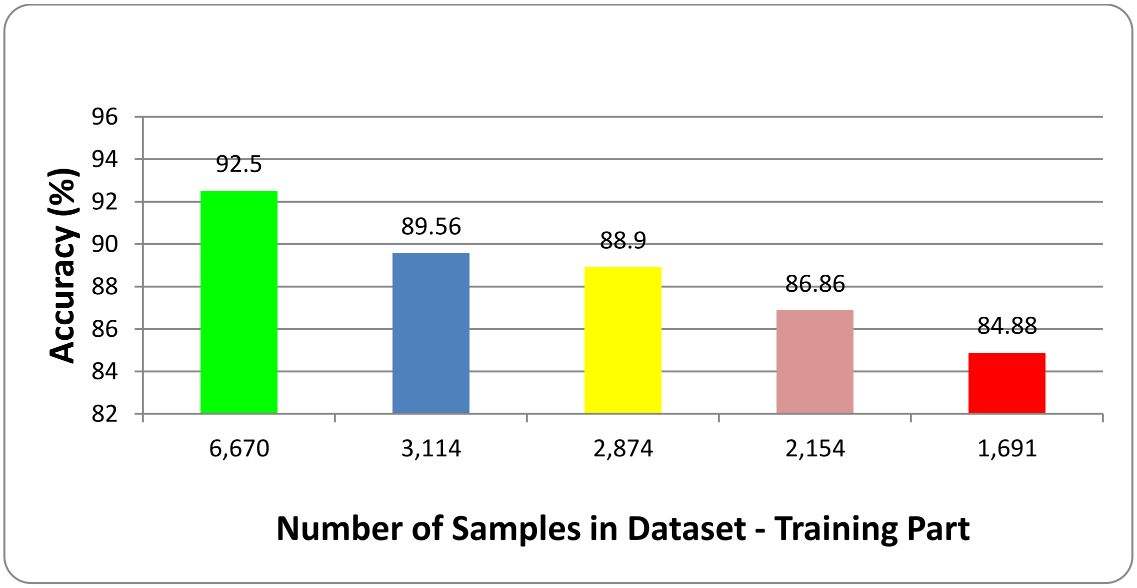


Figure 5.1 : The effect of Vishwanathan's dataset size reduction method on system accuracy, 'OCR' dataset (k-NN classifier)

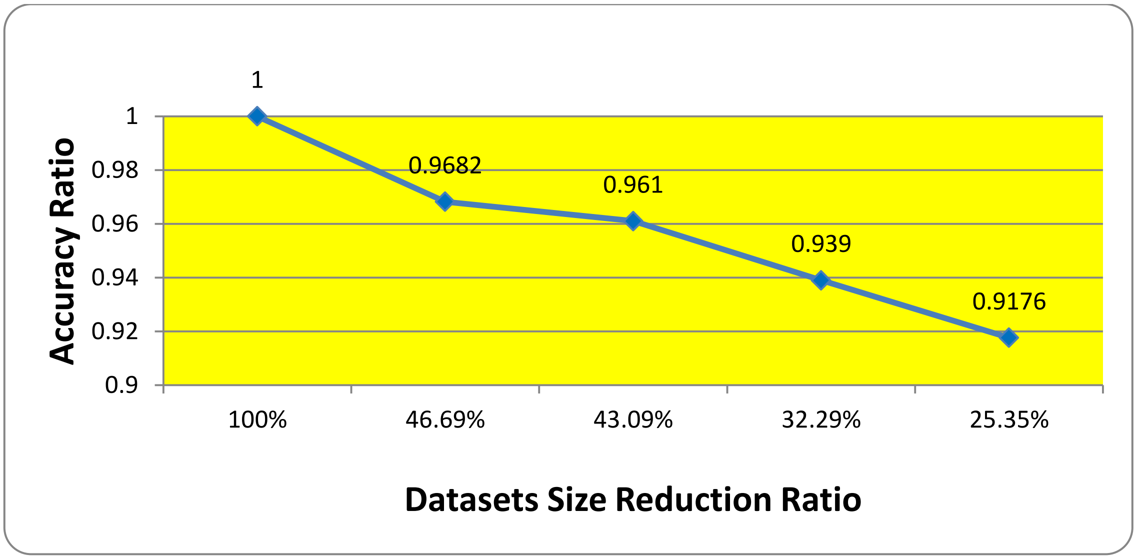


Figure 5.2 : The effect of Vishwanathan's dataset size reduction method on system accuracy, 'OCR' dataset

5.2.1.2 Change Of Classes (COC) Approach

Cervantes, Li and Yu (2008) proposed COC method, as a new version of SVM compared to simple SVM, in order to speeding up the training process of a PR system. They first obtained a sketch from the distribution of available classes with a small number of training samples, and they then identified existing support vector samples 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. They applied their method on IJCNN dataset, including 49,990 training samples and 91,701 testing samples, with 22 numerical features for each record. They decreased the volume of dataset from 100% to 75%, 50%, and 25%, with decreasing system accuracy from initial value 98.5% to 97.9%, 97.4%, and 97.00%, in order (the fifth rows block of Table 5.2). Figure 5.3 shows the accuracy decreasing vs. dataset size reduction, and also, Figure 5.4 shows the accuracy decreasing ratio vs. dataset size reduction ratio for this experiment.

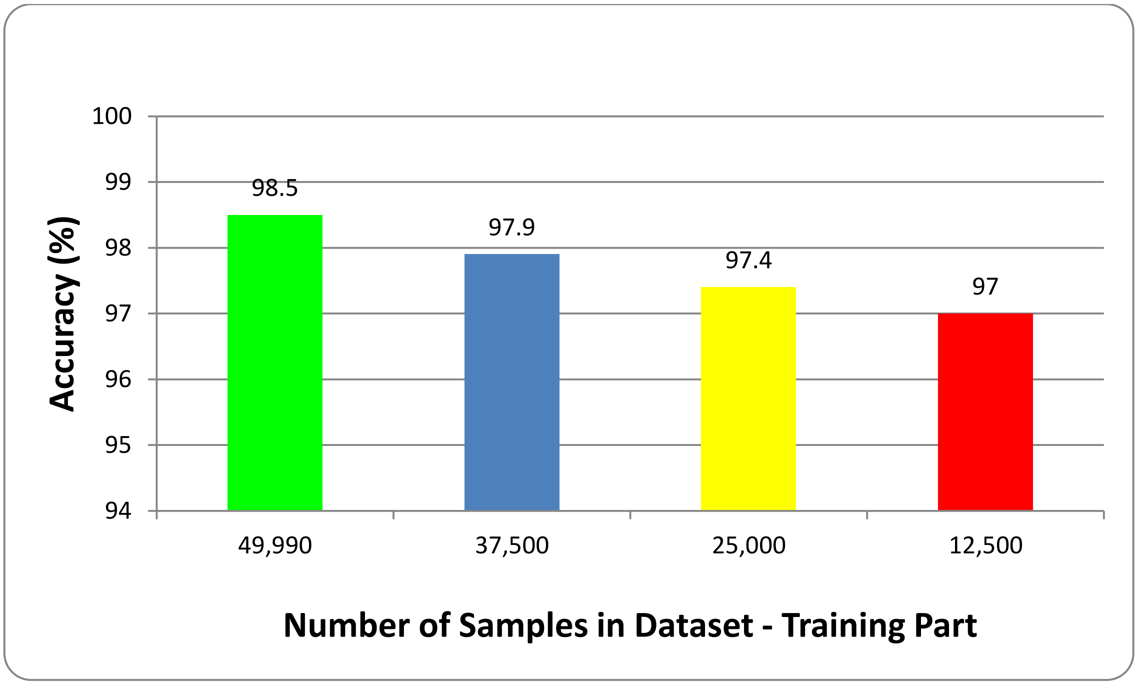


Figure 5.3 : The effect of COC dataset size reduction method on system accuracy, IJCNN dataset (SVM classifier)

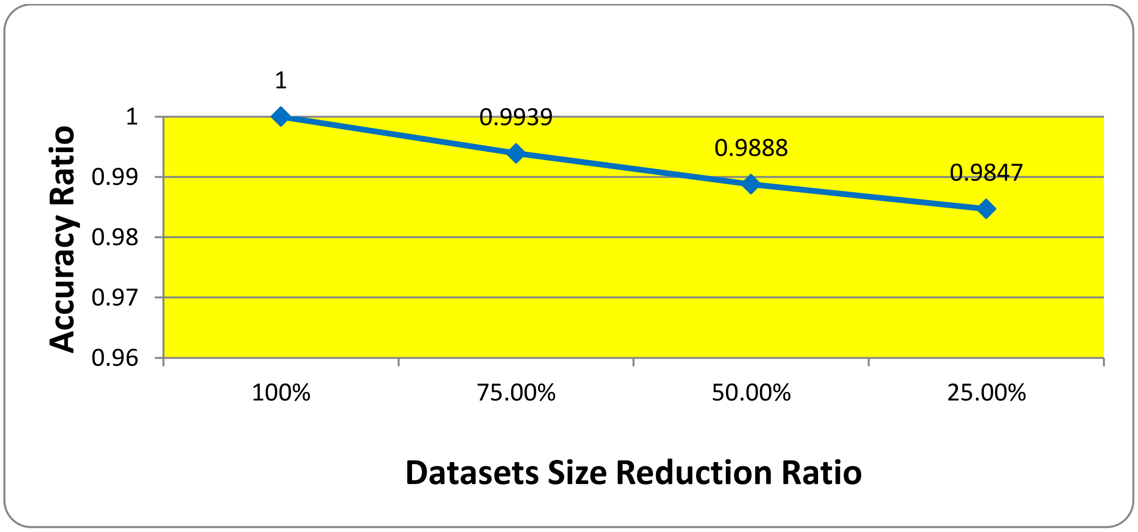


Figure 5.4 : The effect of COC dataset size reduction method on system accuracy, JCNN dataset

5.2.1.3 PA Approach

The Partitioning Approach (PA) for dataset size reduction is another key related work in this domain (Section 2.5.1, and Appendix VI). By applying this approach, the initial dataset volume for digits part of the Hoda dataset was reduced from 100% to 85.12%, 60.84%, 42.88%, and 30.03% for similarity intervals [0.95 – 1], [0.90 - 1], [0.85 - 1], and [0.80 - 1], respectively (Figure 2, Appendix VI). Also, the system accuracy decreased from initial accuracy 96.49% (using all 60,000 training samples) to 96.18%, 95.79%, 95.26%, and 94.62% for similarity intervals [0.95 – 1], [0.9 – 1], [0.85 – 1], and [0.8 – 1], respectively (the second rows block of Table 5.2). Among the generated reduced training datasets, the results show that the best accuracy was obtained when the system was trained with selected samples via similarity interval [0.95-1]. In this case, the dataset volume was reduced by 14.88% (from 100% to 85.12%), but the accuracy slightly decreased only by 0.31% (from 96.49% to 96.18%).

Figure 5.5 and Figure 5.6 show the accuracy decreasing vs. dataset size reduction, and accuracy decreasing ratio vs. dataset size reduction ratio for this experiment, in order. Here, the recognition time for recognizing a sample was also decreased to 87.22%, 62.52%, 44.58%, and 32.57% of the initial recognition time, for similarity intervals [0.95 - 1], [0.90 - 1], [0.85- 1], and [0.80 - 1], respectively.

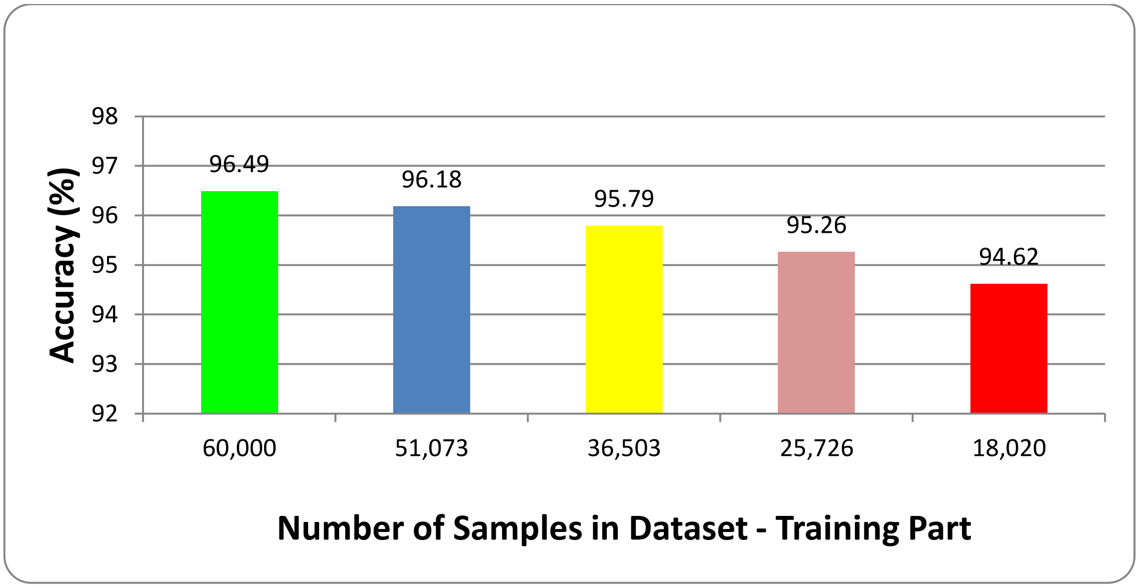


Figure 5.5 : The effect of PA dataset size reduction method on system accuracy – Hoda dataset, digits part (k-NN classifier, k=1)

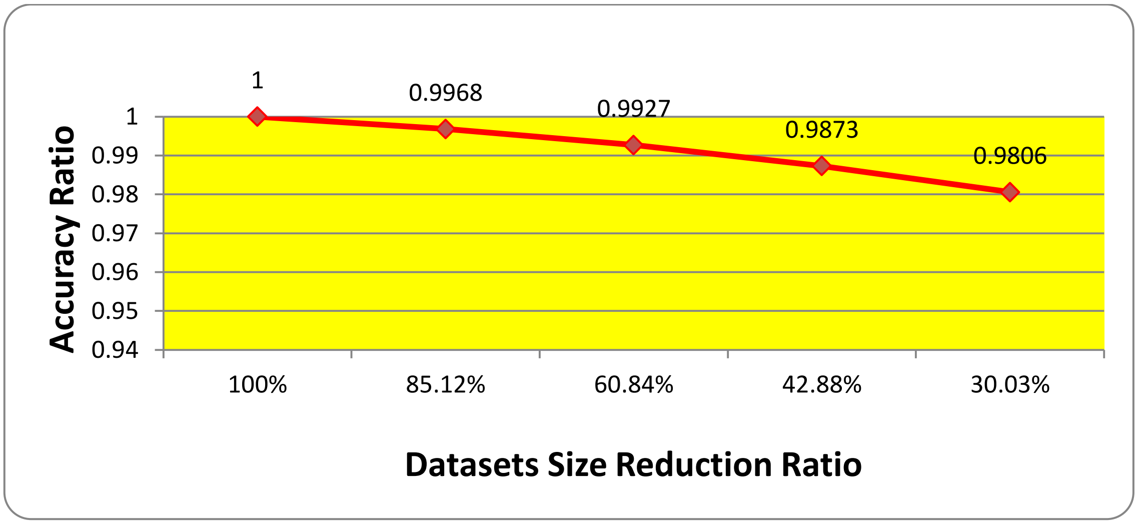


Figure 5.6 : The effect of PA dataset size reduction method on system accuracy – Hoda dataset, digits part

5.2.1.4 The Proposed Method SBR

The proposed dataset size reduction method SBR was explained in Section 4.4. By applying this method on digits part of initial training dataset Hoda, three new reduced versions of training dataset were created: half (30,000 samples) by using sampling rate 1/2, one-third (20,000 samples) by using sampling rate 1/3, and one-fourth (15,000 samples) by using sampling rate 1/4.

Based on the reason that the Vishwanathan approach and PA method used *k*-NN in recognition step, hence, the proposed dataset size reduction method SBR employed a *k*-NN at the recognition stage. Also, to have an accurate comparison with PA method, the same numbers of features, i.e. using 400 image pixels of any image, were used. The obtained accuracies were 96.49%, 95.81%, 95.07%, and 94.78% corresponding to all, 1/2, 1/3, and 1/4 reduced versions of training datasets, respectively (the third rows block in Table 5.2). Figure 5.7 and Figure 5.8 show the accuracy decreasing vs. dataset size reduction, and accuracy decreasing ratio vs. dataset size reduction ratio for this experiment, in order.

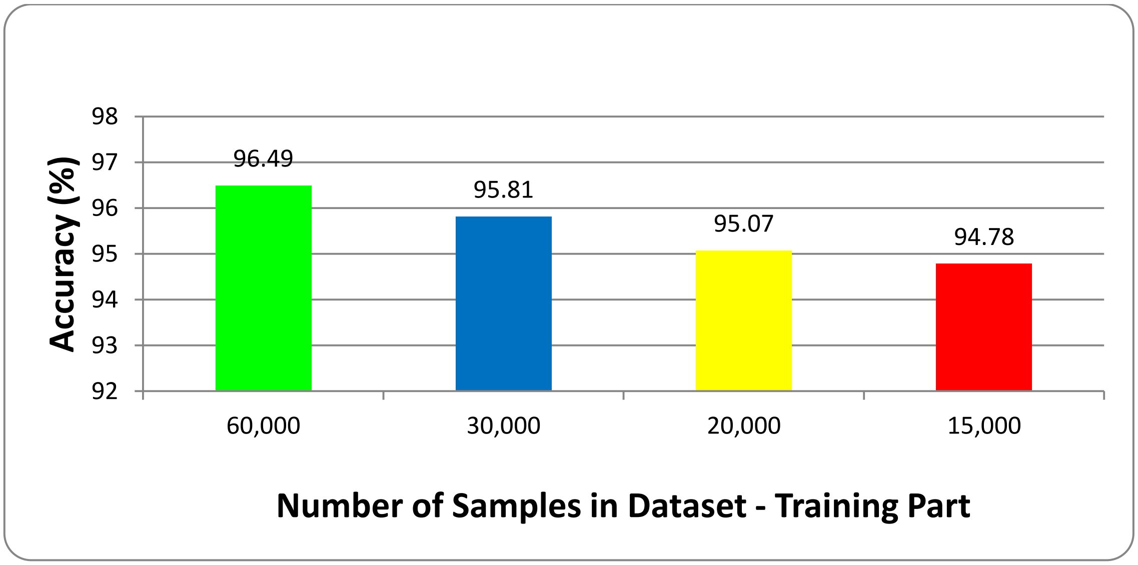


Figure 5.7 : The effect of the proposed SBR dataset size reduction method on system accuracy – Hoda dataset, digits part (k-NN classifier, k=1)

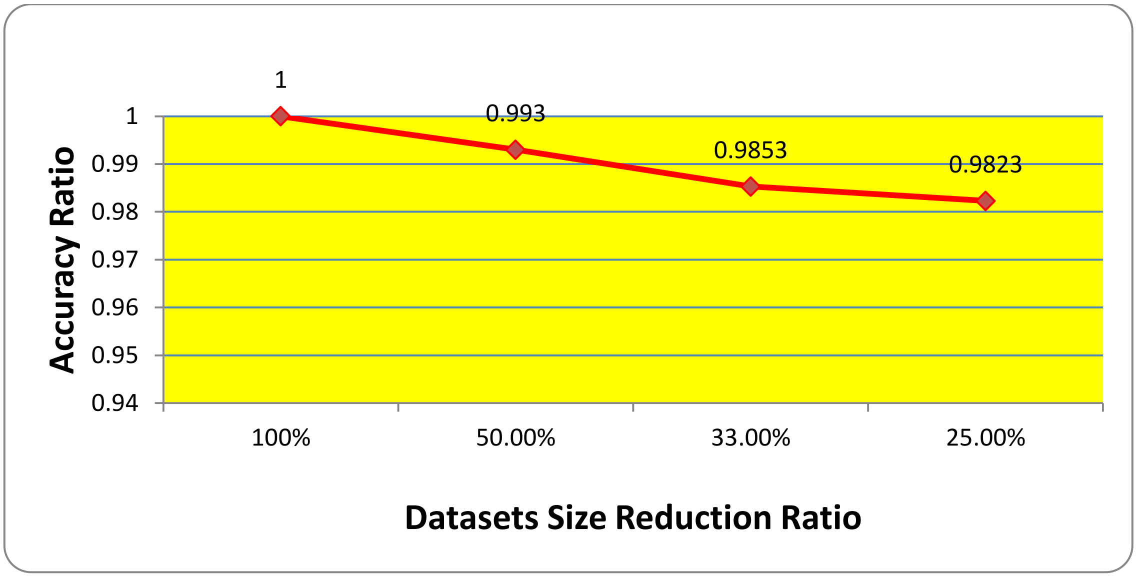


Figure 5.8 : The effect of the proposed SBR dataset size reduction method on system accuracy – Hoda dataset, digits parts

In order to perform more accurate comparison between SBR method and most-related work PA method, some new sub-training datasets were generated and new experiments were carried out.

In PA method, four reduced training dataset including 18,020, 25,726, 36,503, and 51,073 training samples (Table 1 of Appendix VI) were created and used. Hence, the proposed method SBR created again four new subsets of initial Hoda dataset. These new subsets were: $\boldsymbol{N_{1}}$, a 30% (3 out of 10) subset of initial training dataset includes 60,000×30% = 18,000 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.8 – 1]); $\boldsymbol{N_{2}}$, a 42% (21 out of 50) subset of initial training dataset includes 60,000×42% = 25,200 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.85 – 1]); $\boldsymbol{N_{3}}$, a 60% (3 out of 5) subset of initial training dataset includes 60,000×60% = 36,200 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.9 – 1]); and finally ***N*₄,** a 85% (17 out of 20) subset of initial training dataset includes 60,000×85% = 51,000 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.95 – 1]). Table 5.1 compares the number of samples in the reduced training datasets for similarity intervals [0.8 - 1], [0.85 - 1], [0.9 - 1], and [0.95 - 1] introduced by PA method, and the number of samples in the reduced training datasets $\boldsymbol{N_1}$ to $\boldsymbol{N_4}$ proposed by SBR method.

Table 5.1 : Number of samples in the reduced datasets by PA method and reduced datasets by proposed SBR method.

PA Method		The Proposed SBR Method	
Similarity Interval	No. of Samples	$\boldsymbol{N_i}$	No. of Samples
[0.80 - 1]	18,020	$\boldsymbol{N_1}$ (3 out of 10)	18,000
[0.85 - 1]	25,726	$\boldsymbol{N_2}$ (21 out of 50)	25,200
[0.90 - 1]	36,503	$\boldsymbol{N_3}$ (3 out of 5)	36,200
[0.95 - 1]	51,073	$\boldsymbol{N_4}$ (17 out of 20)	51,000

Datasets $\boldsymbol{N_1}$ to $\boldsymbol{N_4}$ were employed as training datasets in following experiments. Similar to PA method, all the 400 pixels of an image were directly fed into the system as features, and a *k*-NN classifier was employed as recognition engine. Here, the initial achieved accuracy using all training samples, without any data size reduction was 96.49% (Section 4.4.5.1). The achieved results are reported in the fourth rows block of Table 5.2. Also, Figure 5.9 compares the achieved accuracy for each of reduced training datasets produced by PA method and reduced datasets $\boldsymbol{N_1}$ to $\boldsymbol{N_4}$ produced by SBR method. The figure clearly shows the superiority of SBR method compared with rival dataset size reduction method PA.

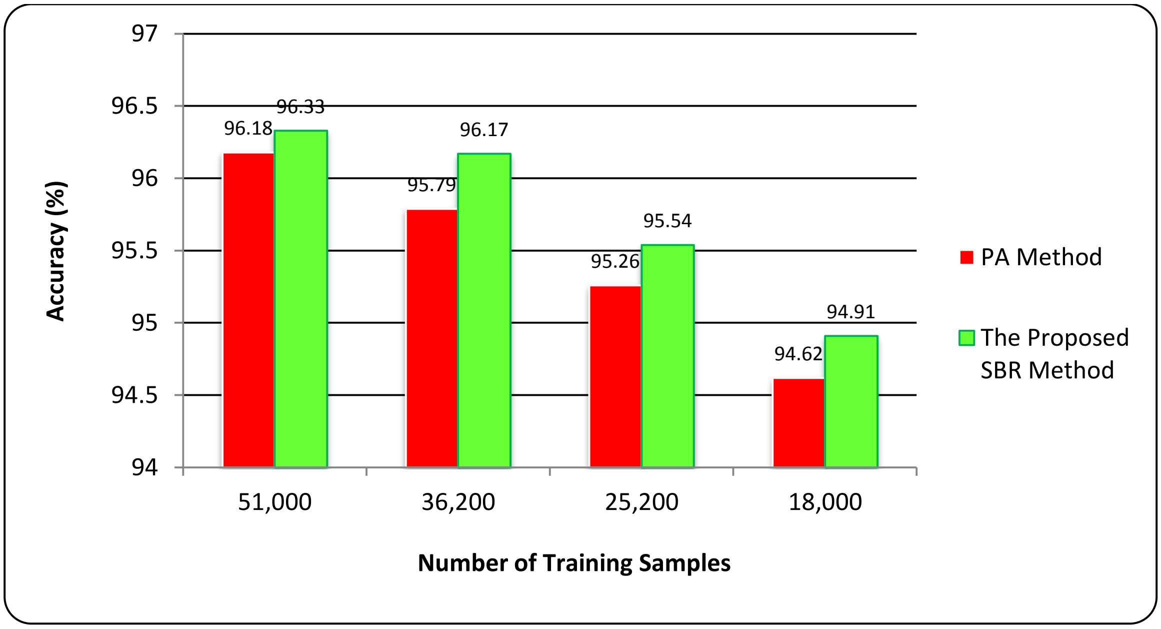


Figure 5.9 : Accuracy comparison between dataset size reduction method PA and proposed dataset size reduction method SBR – Farsi digits recognition (k-NN classifier)

To compare SBR method with COC method, a new reduced version $\boldsymbol{N_5}$ (75% ; 3 out of 4) of original training dataset including 60,000×75% = 45,000 training samples was created, too. Finally, the reduced dataset $\boldsymbol{N_5}$, half (50%), and one-fourth (25%) were employed in a FOCR system, with using SVM classifier (similar to COC method). Here, the initial achieved accuracy using all training samples, without any data size reduction was 98.82%. The accuracy decreased to 98.55%, 98.29%, and 97.94 by using reduced training datasets 75%, 50%, and 25%, in order. The other obtained results are reported in the last rows block of Table 5.2. Also, Figure 5.10 compares the accuracy decreasing ratio of COC method with accuracy decreasing ratio of SBR method. The figure shows the superiority of the proposed SBR method against dataset size reduction method COC for all dataset reduction schemes.

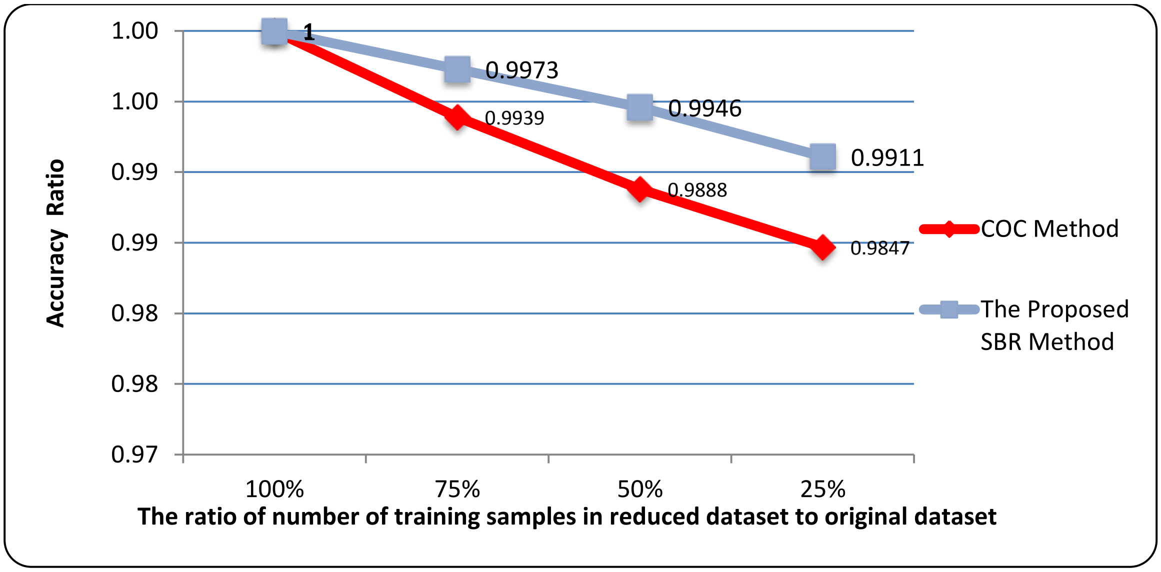


Figure 5.10 : Accuracy comparison between dataset size reduction method COC and proposed dataset size reduction method SBR – Farsi digits recognition

Table 5.2 : The results of various dataset size reduction approaches – Farsi digits

References		Dataset	Different versions of dataset	No. of samples in reduced dataset version	The ratio of reduced dataset volume to initial dataset volume	Number of Features	Classifier	Final Accuracy %	The ratio of the new accuracy to the initial accuracy
1	Vishwanathan's method (2004)	OCR	Original Dataset	6,670	-----------	----	*k*-NN	92.50	1
			-------------	3,114	46.69%	----		89.56	0.9682
			-------------	2,874	43.09%	----		88.90	0.9610
			-------------	2,154	32.29%	----		86.86	0.9390
			-------------	1,691	25.35%	----		84.88	0.9176
2	PA method (Shayegan & Aghabozorgi, 2014)	Hoda	Original Dataset	60,000	--------	400	*k*-NN	96.49	1
			[0.95 – 1]	51,073	85.12%	400		96.18	0.9968
			[0.90 – 1]	36,503	60.84%	400		95.79	0.9927
			[0.85 – 1]	25,726	42.88%	400		95.26	0.9873
			[0.80 – 1]	18,020	30.03%	400		94.62	0.9806
3	The Proposed SBR Method Experiments, #1	Hoda	Original Dataset	60,000	---------	400	*k*-NN	96.49	1
			1/2 version	30,000	50%	400		95.81	0.9930
			1/3 version	20,000	33%	400		95.07	0.9853
			1/4 version	15,000	25%	400		94.78	0.9823
4	The Proposed SBR Method Experiments, #2	Hoda	Original Dataset	60,000	--------	400	*k*-NN	96.49	1
			N4	51,000	85%	400		96.33	0.9983
			N3	36,200	60%	400		96.17	0.9967
			N2	25,200	42%	400		95.54	0.9902
			N1	18,000	30%	400		94.91	0.9836
5	COC method (Cervantes et al., 2008)	IJCNN	Original Dataset	49,990	----------	22	SVM	98.5%	1
			-------------	37,500	75%	22		97.9%	0.9939
			-------------	25,000	50%	22		97.4%	0.9888
			-------------	12,500	25%	22		97.0%	0.9847
6	The Proposed SBR Method Experiments, #3	Hoda	Original Dataset	60,000	--------	400	SVM	98.82	1
			N5 : 3/4 version	45,000	75%	400		98.55	0.9973
			1/2 version	30,000	50%	400		98.29	0.9946
			1/4 version	15,000	25%	400		97.94	0.9911

5.2.1.5 Discussion

Vishwanathan's method and PA method used *k*-NN classifier, and COC method used SVM classifier. Hence, the proposed SBR method was utilized with employing both classifiers *k*-NN and SVM, to perform more accurate comparison with the related works. The Hoda dataset includes about 10 times more training samples compared to the 'OCR' dataset, used in Vishwanathan's method. Also, the Hoda dataset includes 20% more training samples compared to IJCNN dataset, used in COC method.

The Vishwanathan's method reduced the volume of the training samples from 100% to 43.09%, but it decreased the accuracy to 0.9610 of initial value, from 92.50% to 88.90% (the first yellow colored row in Table 5.2). The COC method reduced the volume of the training samples from 100% to 50% and 25%, but it decreased the accuracy from 98.50% to 97.40% and 97.0% in order (the light blue colored rows in Table 5.2). PA method succeeded to reduce the volume of training samples from 100% to 42.88% (almost similar to Vishwanathan's method), while the accuracy decreased to 0.9873 of the initial value, from 96.49% to 95.26% (second yellow colored row in Table 5.2).

a) Comparison between SBR method and COC and Vishwanathan methods:

In experiment #1, the proposed SBR method succeeded in decreasing the volume of training samples from 100% to 33.00% and from 100% to 25% (almost similar to third and fourth rows of Vishwanathan's method), while the accuracy decreased to 0.9853 and 0.9823 of the initial value (from 96.49% to 95.07%, and from 96.49% to 94.78%, the first and second green colored rows in Table 5.2). Here, both of the achieved results were better than the obtained results by Vishwanathan's method. Also, in this experiment, the proposed SBR method succeeded in decreasing the volume of training samples from 100% to 50%

and from 100% to 25% (almost similar to third and fourth rows of COC method), while the accuracy decreased to 0.9930 and 0.9823 of the initial value (from 96.49% to 95.81%, and from 96.49% to 94.78%, the first and third green colored rows in Table 5.2).

b) Comparison between SBR method and PA method:

In experiment #2, the proposed SBR method succeeded in decreasing the volume of training samples from 100% to 85%, 60%, 42%, and 25%, (almost similar to PA method), while the accuracy decreased to 0.9983, 0.9967, 0.9902, and 0.9836 of the initial value, in order. Here, all the results were higher than the obtained results by PA method in similar conditions.

c) Comparison between SBR method and COC method:

In experiment #3, the proposed SBR method decreased the volume of training samples from 100% to 75%, 50%, and 25%, (similar to COC method), while the accuracy ratio decreased to 0.9973, 0.9946, and 0.9911 of the initial value, in order (the last block rows of Table 5.2). Here, the results were higher than the obtained results by COC method, for all reduced datasets.

Although the conditions of these experiments are not exactly similar, but it is still evident that the PA method outperformed Vishwanathan’s method, and SBR method outperformed PA method, in terms of recognition accuracy. Also, SBR method, with using *k*-NN classifier, shows better accuracy compared to COC method, till reduction rate 33%. For reduction rate more than 33%, COC method creates the better accuracy compared to SBR method, using *k*-NN. Finally, SBR method with using SVM classifier (Experiment 3#) achieved to higher accuracy for all reduced training datasets.

To find the main reason of outperforming of SBR method against the most related work PA method, the misclassified samples in both experiments were investigated. The majority of misrecognized samples belonged to degraded and low quality samples, which their initial shape were far from corresponding class templates. These samples are put in the second partition proposed by PA method. Hence, they were saved in the final reduced training dataset. But, SBR method created a reduced dataset that any two successive training samples (based on their similarity values) were more separated from each other. This characteristic helped the recognition engine *k*-NN to classify the input instance more accurate. Figure 5.11 plots the ratio of accuracy decreasing vs. the ratio of dataset size reduction, corresponding to results in Table 5.2. It is clearly shows the superiority proposed SBR method compared to the literature.

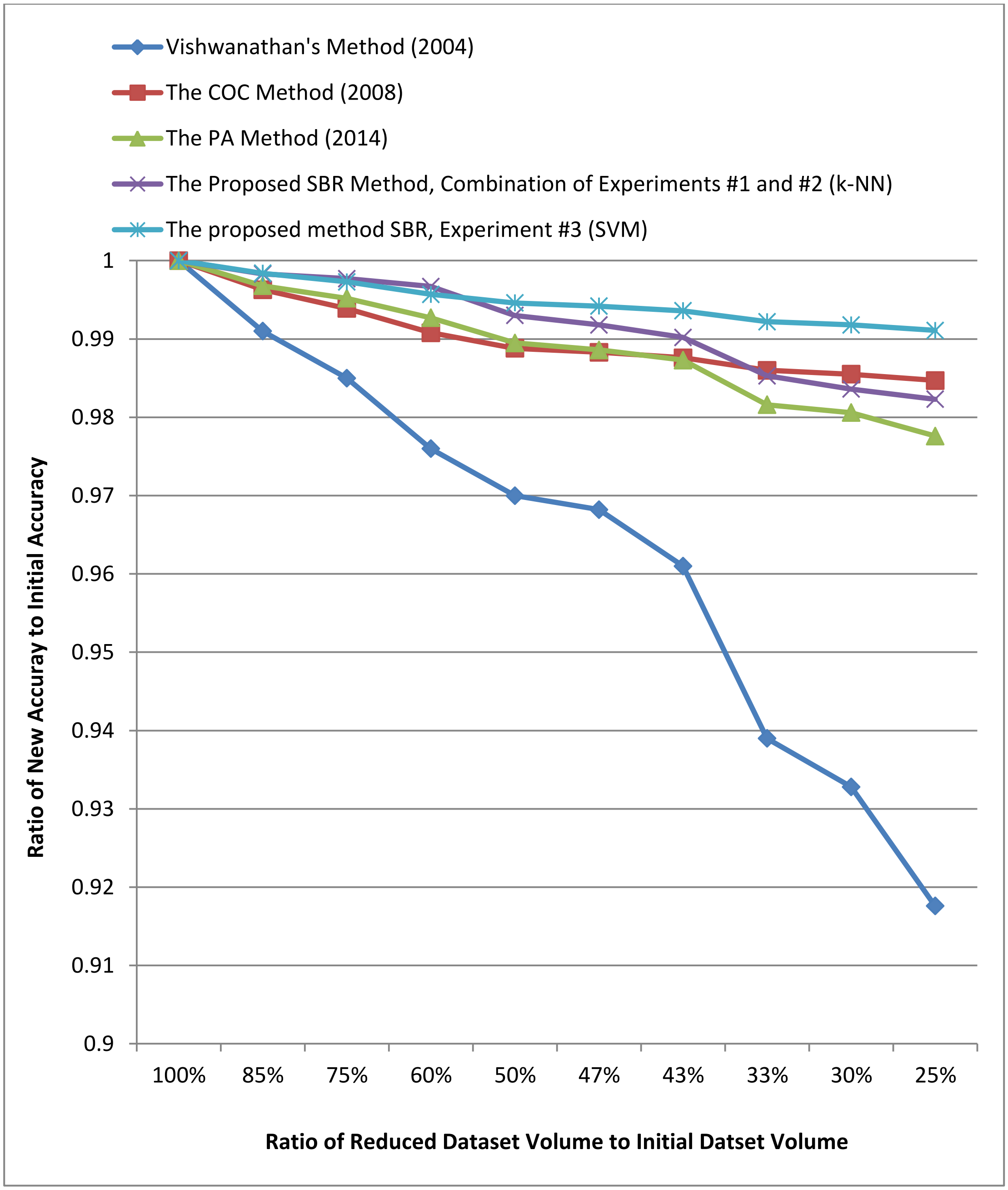


Figure 5.11 : Accuracy decreasing vs. dataset size reduction, corresponding to Table 5.2

5.2.2 Letters Samples

5.2.2.1 PA Method

Shayegan and Aghabozorgi (2014b) decreased the number of training samples of the characters part of Hoda dataset from the initial volume 100% to 91.66%, 79.32%, 65.08%, and 48.54% for similarity intervals [0.95 – 1], [0.90 - 1], [0.85 - 1], and [0.80 - 1], respectively (Section 2.5.1, and Table 3 of Appendix VI). Also, the system accuracy decreased from initial accuracy 80.67% (using all 70,645 training samples) to 80.39%, 78.47%, 72.48%, and 64.13% for similarity intervals [0.95 – 1], [0.9 – 1], [0.85 – 1], and [0.8 – 1], respectively (Figure 3 of Appendix VI). The recognition time for a sample decreased to 92.39%, 81.69%, 67.12% and 50.74% of the initial recognition time, for similarity intervals [0.95 - 1], [0.90 - 1], [0.85 - 1], and [0.80 - 1], respectively (Table 4 of Appendix VI). The first rows block of Table 5.4 shows the related results. In this experiment, the results showed that the best accuracy was obtained when the system was trained with selected samples via similarity interval [0.95-1]. In this case, the dataset volume was reduced by 8.34% (from 100% to 91.66%), but the accuracy slightly decreased only by 0.28% (from 80.67% to 80.39%). Also, the results showed that the proposed FOCR system achieved higher accuracy, when it was trained by the reduced versions of initial dataset ***R_Dataset***, proposed by PA method, in comparison with the system when it was trained by the other subset of the initial dataset (subsets **S_1, S_2, S_3,** and **S_4**) with the same size (number of training samples) (Figure 3 of Appendix VI).

Figure 5.12 and Figure 5.13 show the accuracy decreasing vs. dataset size reduction, and accuracy decreasing ratio vs. dataset size reduction ratio for this experiment, in order. Here, the recognition time for recognizing a sample was also decreased to 92.39%, 81.69%, 67.12%, and 50.74% of the initial recognition time, for similarity intervals [0.95 - 1], [0.90 - 1], [0.85- 1], and [0.80 - 1], respectively.

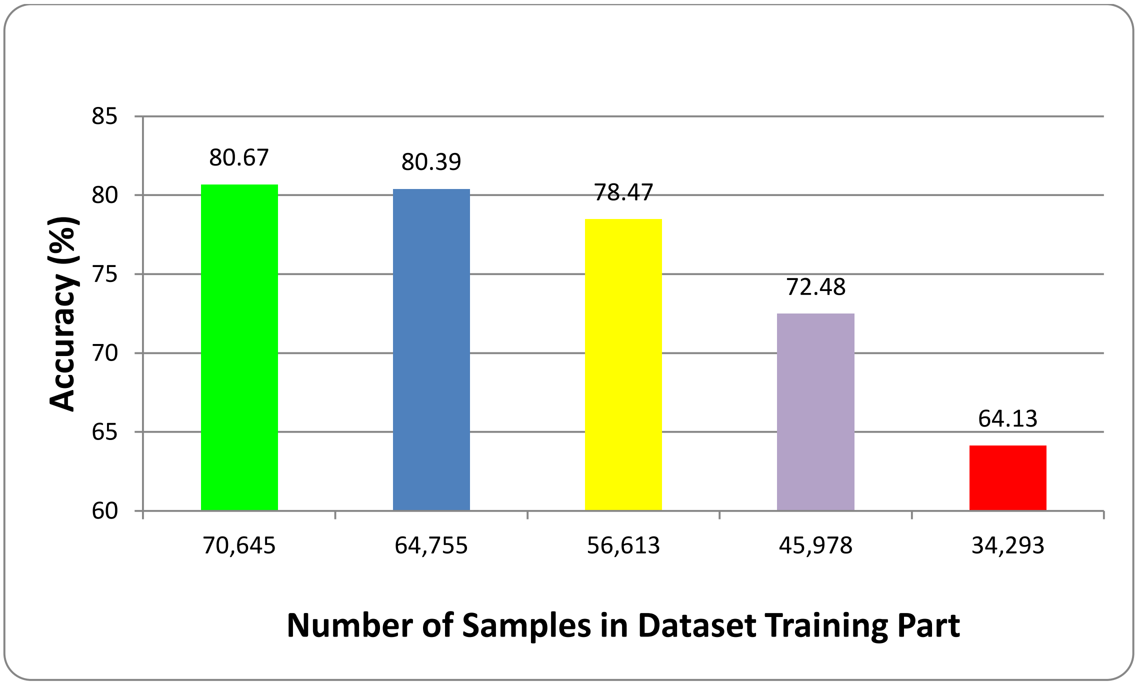


Figure 5.12 : The effect of PA dataset size reduction method on system accuracy - Hoda dataset, characters part (k-NN classifier, k=1)

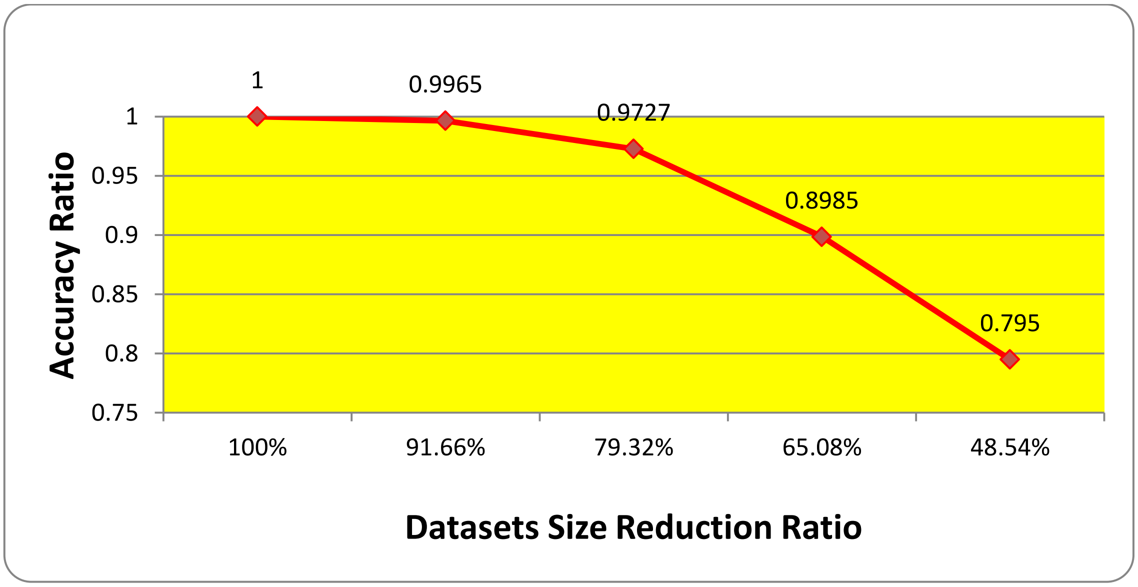


Figure 5.13 : The effect of PA dataset size reduction method on system accuracy – Hoda dataset, characters part

No other research works, regarding dataset size reduction for Farsi letters, were found to compare them to the proposed dataset size reduction SBR method. However, from an OCR view of point, the obtained results, by the proposed FOCR system, are compared to the literature in Section 5.4.2.

5.2.2.2 The Proposed Method SBR

By applying dataset size reduction method SBR, three new reduced versions of training dataset (for characters part of the Hoda dataset) were created (Section 4.4.5.2): half (35,322 samples) by using sampling rate 1/2, one-third (23,548 samples) by using sampling rate 1/3, and one-fourth (17,661 samples) by using sampling rate 1/4. These reduced datasets along with the initial dataset were used in different experiments.

The PA method used *k*-NN in recognition phase. Hence, the proposed dataset size reduction method SBR employed a *k*-NN at the recognition stage, too. Also, to do a more accurate comparison with PA method, the same numbers of features, i.e. using 400 image pixels of any image, were used. The obtained accuracies were 80.67%, 79.36%, 74.64%, and 69.46% corresponding to all, 1/2, 1/3, and 1/4 reduced versions of training datasets, respectively. Figure 5.14 and Figure 5.15 show the accuracy decreasing vs. dataset size reduction, and accuracy decreasing ratio vs. dataset size reduction ratio for this experiment, in order. Also, the achieved results have been shown in the second rows block in Table 5.4.

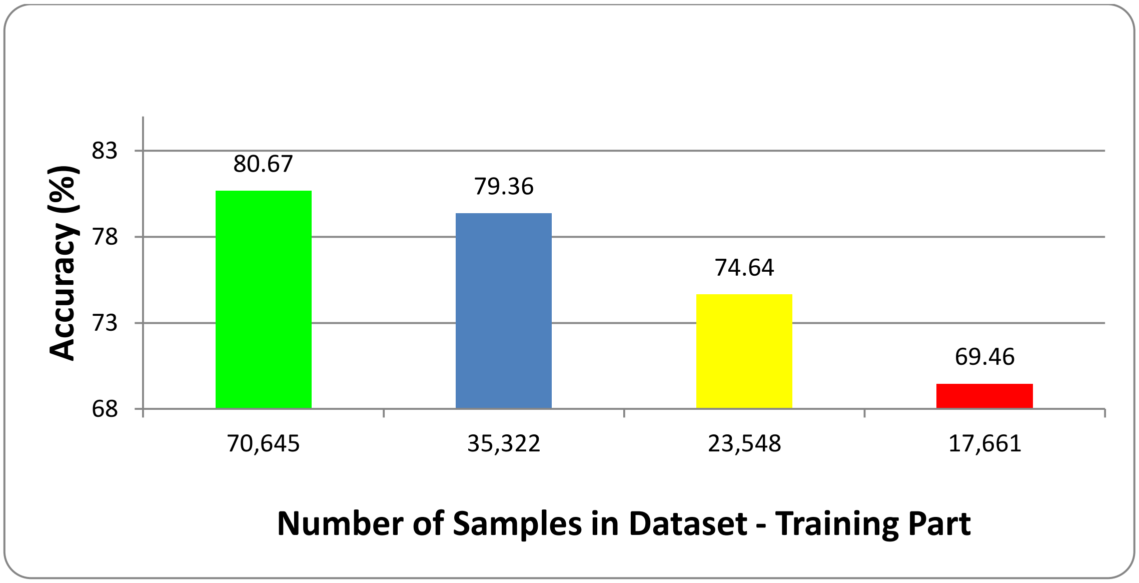


Figure 5.14 : The effect of the proposed SBR dataset size reduction method on system accuracy – Hoda dataset, characters part (k-NN classifier, k=1)

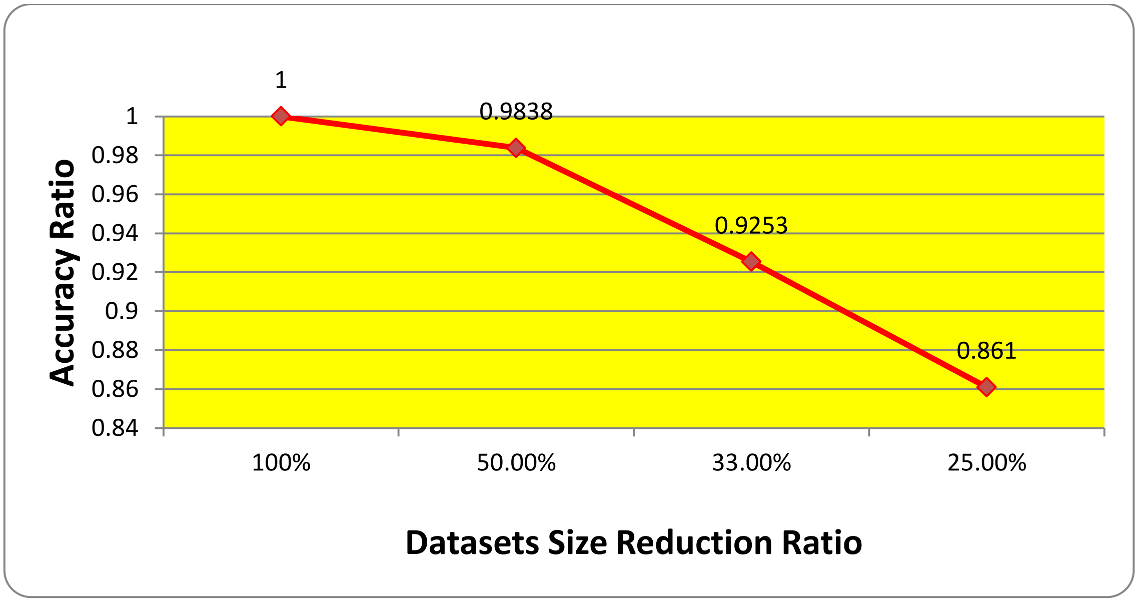


Figure 5.15 : The effect of the proposed SBR dataset size reduction method on system accuracy – Hoda dataset, characters part

In order to perform an accurate comparison between SBR method and PA method, some new sub-training datasets were generated and new experiments were carried out.

In PA method, four reduced training dataset including 34,293, 45,978, 56,613, and 64,755 training samples (Table 3 of Appendix VI) were created and used. Hence, four new subset of initial Hoda dataset were created again, by using the proposed dataset size reduction SBR method. These new subsets were: ***N_1**,* a 48% (12 out of 25) subset of initial training dataset includes 70,645×48% = 33,909 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.8 – 1]); ***N_2**,* a 65% (13 out of 20) subset of initial training dataset includes 70,645×65% = 45,919 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.85 – 1]); ***N_3**,* a 80% (4 out of 5) subset of initial training dataset includes 70,645×80% = 56,516 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.9 – 1]); and ***N_4**,* a 92% (23 out of 25) subset of initial training dataset includes 70,645×92% = 64,993 training samples (which is almost equal to the number of samples in the subset created using similarity interval [0.95 – 1]). Table 5.3 compares the number of samples in the reduced training datasets for similarity intervals [0.8 - 1], [0.85 - 1], [0.9 - 1], and [0.95 - 1] introduced by PA method, and the number of samples in the reduced training datasets ***N_1*** to ***N_4*** proposed by SBR method.

Table 5.3 : Number of samples in the reduced datasets by PA method and reduced datasets by proposed SBR method

PA Method		The Proposed SBR Method	
Similarity Interval	No. of Samples	***N_i***	No. of Samples
[0.80 - 1]	34,293	***N_1*** (12 out of 25)	33,909
[0.85 - 1]	45,978	***N_2*** (13 out of 20)	45,919
[0.90 - 1]	56,613	***N_3*** (4 out of 5)	56,516
[0.95 - 1]	64,755	***N_4*** (23 out of 25)	64,993

The datasets ***N_1*** to ***N_4*** were employed as training datasets, similar to PA method, and all the 400 pixels of an image were directly fed into the system as features, and a *k*-NN classifier was employed as recognition engine. Here, the initial achieved accuracy using all training samples, without any data reduction was 80.67% (Section 4.4.5.2). The achieved results are reported in the third rows block of Table 5.4. Figure 5.16 compares the achieved accuracy for each of reduced training datasets produced by PA method and reduced datasets ***N_1*** to ***N_4*** produced by SBR method. This figure shows the superiority of reduced datasets ***N_1*** to ***N_4*** compared with rival reduced datasets, created by PA method, in training a FOCR system.

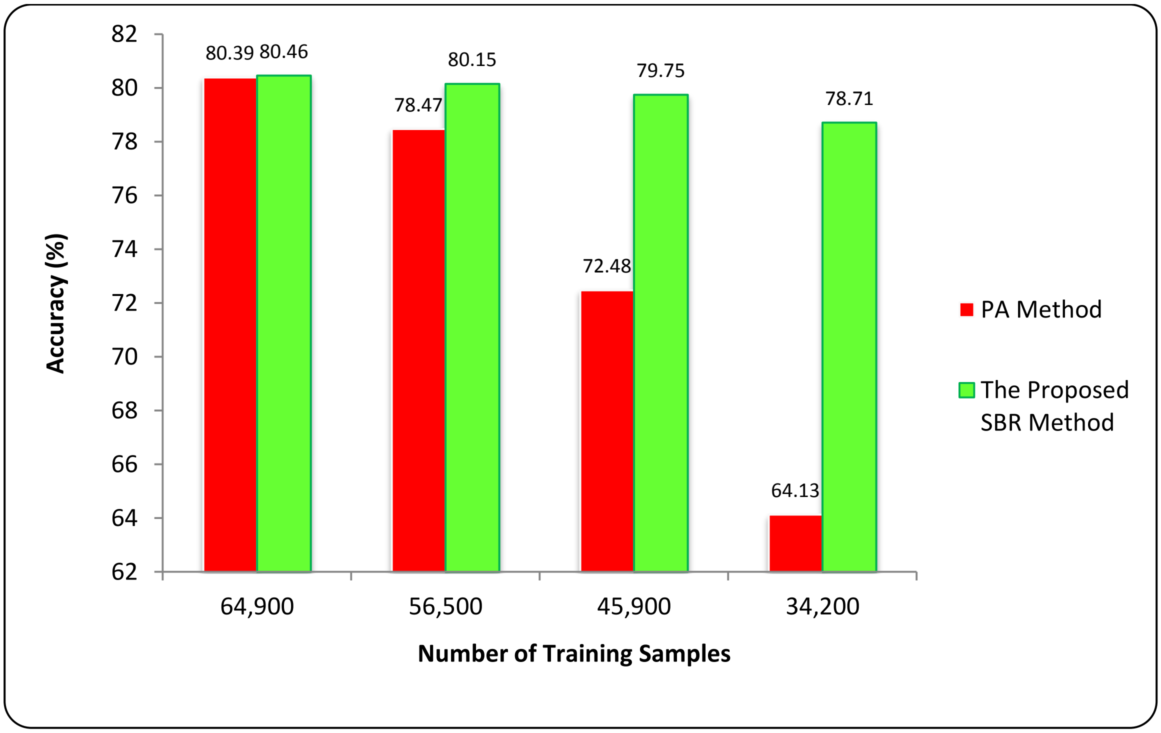


Figure 5.16 : Accuracy comparison between dataset size reduction method PA and proposed dataset size reduction method SBR – Farsi characters recognition

Table 5.4 : The results of various dataset size reduction approaches – Farsi letters

References		Dataset	Different versions of datasets	No. of samples in reduced dataset version	The ratio of reduced dataset volume to initial dataset volume	Number of Features	Final Accuracy %	The ratio of the new accuracy to the initial accuracy
1	PA method (Shayegan & Aghabozorgi, 2014)	Hoda	Original Dataset	70,645	--------	400	80.67	1
			[0.95 – 1]	64,755	91.66%	400	80.39	0.9965
			[0.90 – 1]	56,613	79.32%	400	78.47	0.9727
			[0.85 – 1]	45,978	65.08%	400	72.48	0.8985
			[0.80 – 1]	34,293	48.54%	400	64.13	0.7950
2	The Proposed SBR Method Experiment #1	Hoda	Original Dataset	70,645	---------	400	80.67	1
			1/2 version	35.322	50%	400	79.36	0.9838
			1/3 version	23,548	33%	400	74.64	0.9253
			1/4 version	17,661	25%	400	69.46	0.8610
3	The Proposed SBR Method Experiment #2	Hoda	Original Dataset	70,645	--------	400	80.67	1
			N4	64,993	92%	400	80.46	0.9974
			N3	56,516	80%	400	80.15	0.9936
			N2	45,919	65%	400	79.75	0.9886
			N1	33,909	48%	400	78.71	0.9757

5.2.2.3 Discussion

The PA method succeeded to reduce the volume of training samples from 100% to 48.54%, while the accuracy decreased dramatically to 0.7950 of the initial value, from 80.67% to 64.13%. In experiment #1, the proposed SBR method succeeded in decreasing the volume of training samples from 100% to 50% (almost similar to the first green colored row in Table 5.4), while the accuracy decreased to 0.9838 of the initial value (from 80.67% to 79.36%, the second green colored row in Table 5.4). This result is very better than obtained result in PA method. In experiment #2, the proposed SBR method succeeded in decreasing the volume of training samples from 100% to 92%, 80%, 65%, and 48%, (almost similar to PA method), while the accuracy decreased to 0.9974, 0.9936, 0.9886, and 0.9757 of the initial value, in order (the third rows block of Table 5.4). Here, all the accuracies were higher than the obtained accuracies by PA method in similar conditions.

Although the conditions of these experiments are not exactly similar, (there are a small difference between the number of training samples), but it is still evident that the SBR method outperformed PA method, in terms of recognition accuracy. Figure 5.17 plots the ratio of accuracy decreasing vs. the ratio of dataset size reduction, corresponding to results in Table 5.4.

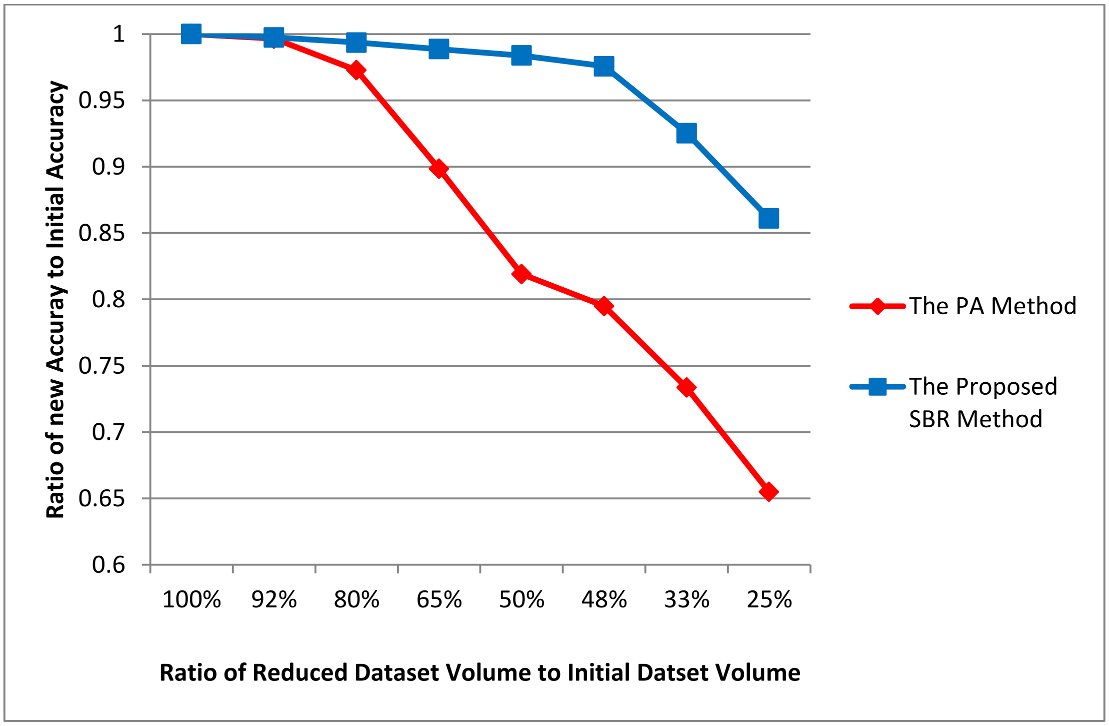


Figure 5.17 : Accuracy decreasing vs. dataset size reduction, corresponding to Table 5.4

5.3 Results Comparison for Dimensionality Reduction Operation

To compare the performance of the proposed dimensionality reduction method 2S_SA against other well-known feature selection techniques, a general PCA technique (Appendix III, and Section 2.3.3.2.c) and a Random Projection (RP) technique (Appendix V, and

Section 2.3.3.2.c) were applied on the initial features vector ***Initial_S***. The following sub-sections demonstrate the related experiments.

5.3.1 Dimensionality Reduction Operation in Farsi Digits Recognition

In the case of recognizing Farsi digits, a FOCR system was first trained with using 133-dimensional features vector ***Initial_S*** (Section 4.6.2.1) and it achieved to 90.41% accuracy. Then, the system was trained with the proposed final version of features vector ***F/D-S2*** (generating by the proposed 2S_SA method) with only 58 features (Section 4.6.2.1), and the correct recognition rate increased from 90.41% to 95.12%. To find the superiority of the features set ***F/D-S2***, compared to the other subsets of ***Initial_S***, which have 58 members, the following experiments were carried out. The PCA technique was applied on initial features vector ***Initial_S***, and it created another features set ***Temp*** in the new orthogonal features space, based on the derived eigenvectors. The first 58 elements of features vector ***Temp*** created a new small features vector ***F/D-PCA***. This features set was fed into the same employed MLP-NN classifier Section 4.6.2.1, and a final accuracy of 89.00% was achieved. Similarly, the dimensionality reduction technique RP was employed to create a new smaller version ***F/D-RP*** with 58 elements of initial features vector ***Initial_S***. The features vector ***F/D-RP*** was fed into the same employed MLP-NN classifier Section 4.6.2.1. Here, the recognition rate declined dramatically to 83.66%. The outcome results are reported in the first row-block of Table 5.5.

5.3.2 Dimensionality Reduction operation in Farsi Characters Recognition

In the case of recognizing Farsi characters, an OCR system was first trained with using 133-dimensional features vector ***Initial_S*** (Section 4.6.2.2) and it achieved to 81.82% accuracy. Then, the system was trained with the proposed final version of features vector

F/C-S2 (generating by the proposed 2S_SA method) with only 93 features (Section 4.6.2.2). On average, the correct recognition rate increased from 81.82% to 83.74%. To find the superiority of the features set ***F/C-S2*** compared to the other subsets of ***Initial_S***, which have 93 members, the following experiments were carried out. The PCA technique was applied on initial features vector ***Initial_S***, and it created another features set ***Temp*** in the new orthogonal features space, based on the derived eigenvectors. The first 93 elements of features vector ***Temp*** created a new small features vector ***F/C-PCA***. This features set was fed into the same employed MLP-NN classifier Section 4.6.2.2, and a final accuracy of 80.27% was achieved. Similarly, the dimensionality reduction technique RP was employed to create a new smaller version ***F/C-RP*** with 93 elements of initial features vector ***Initial_S***. The features vector ***F/C-RP*** was fed into the same employed MLP-NN classifier Section 4.6.2.2. Here, the recognition rate declined dramatically to 75.09%. The outcome results are reported in the second row-block of Table 5.5.

5.3.3 Dimensionality Reduction operation in English Digits Recognition

For English digits recognition, the trend was completely similar to previous experiments for Farsi digits and letters. In this case, the OCR system was first trained with using 133-dimensional features vector ***Initial_S*** (Section 4.6.2.3) and it achieved to 91.93% accuracy. Then, the system was trained with the proposed final version of features vector ***E-S2*** (generating by the proposed 2S_SA method) with only 79 features (Section 4.6.2.3). On average, the correct recognition rate increased from 91.93% to 94.88%. To find the superiority of the features set ***E-S2***, compared to the other subsets of ***Initial_S***, which have 79 members, the following experiments were carried out. The PCA technique was applied on initial features vector ***Initial_S***, and it created another features set ***Temp*** in the new orthogonal features space, based on the derived eigenvectors. The first 79 elements of features vector ***Temp*** created a new smaller features vector ***E-PCA***. This features set was fed into the same employed MLP-NN classifier Section 4.6.2.3, and a final accuracy of 90.71% was achieved. Similarly, the dimensionality reduction technique RP was employed to create a new smaller version ***E-RP*** with 79 elements of initial features vector ***Initial_S***. The features vector ***E-RP*** was fed into the same employed MLP-NN classifier Section 4.6.2.3. Here, the recognition rate decreased to 88.39%. The obtained results are reported in the third row-block of Table 5.5.

Table 5.5 : Comparison between the proposed dimensionality reduction technique 2S_SA with PCA and RP techniques (ANN classifier)

Dataset		Feature Set	Number of Features in Feature Vector	Feature Selection Method			Accuracy %
				2S_SA	PCA	RP	
1	Farsi Dataset Hoda Digits Part	Initial_S	133	---------------			90.41
		F/D-S2	58	*			95.12
		F/D-PCA	58		*		89.00
		F/D-RP	58			*	83.66
2	Farsi Dataset Hoda Characters Part	Initial_S	133	--------------			81.82
		F/C-S2	93	*			83.74
		F/C-PCA	93		*		80.27
		F/C-RP	93			*	75.09
3	English dataset MNIST	Initial_S	133	---------------			91.93
		E-S2	79	*			94.88
		E-PCA	79		*		90.71
		E-RP	79			*	88.39

5.3.4 Discussion

All the obtained results showed the superiority of the proposed dimensionality reduction method 2S_SA compared to PCA and RP, as two well-known dimensionality reduction techniques (the coloured green rows in Table 5.5). For Farsi digits recognition, the achieved accuracy with using the proposed technique 2S_SA is 6.12% and 11.46% higher than the achieved accuracy with PCA and RP techniques, in order. For Farsi letters recognition, the achieved accuracy with using the proposed technique 2S_SA is 3.47% and 8.65% higher than the achieved accuracy with PCA and RP techniques, in order. For English digits recognition, the achieved accuracy with using the proposed technique 2S_SA is 4.17% and 6.49% higher than the achieved accuracy with PCA and RP techniques, in order.

In these experiments, the PCA technique outperformed to RP technique for dimensionality reduction purpose. However, it is worth mentioning that some researchers have shown the superiority of RP technique against PCA technique, for dimension reduction purpose, in high-dimensional features space condition (Deegalla & Bostrm, 2006; Fradkin & Madigan, 2003). Figure 5.18 shows a comparison between the traditional feature selection techniques PCA and RP, and the proposed feature selection 2S_SA for Farsi digits, Farsi letters, and English digits recognition systems. It indicates that the recognition systems achieved to higher recognition rate when the proposed feature selection 2S_SA was employed.

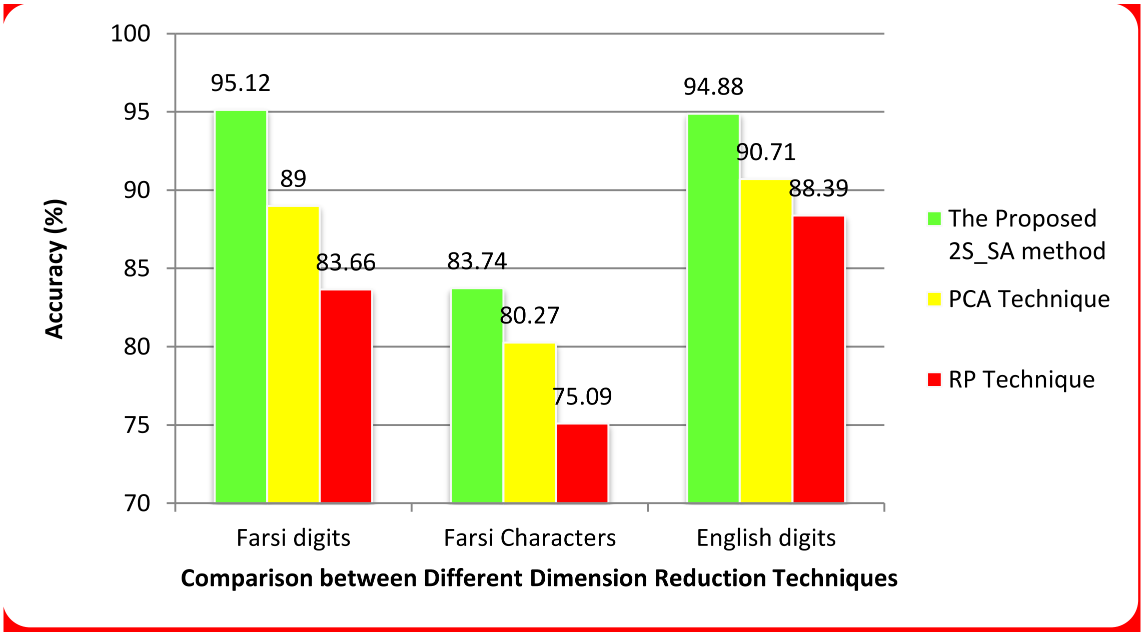


Figure 5.18 : Accuracy comparison between traditional feature selection methods PCA and RP with the proposed dimensionality reduction method 2S_SA

In order to perform an accurate comparison between proposed method 2S_SA and powerful dimensionality reduction technique PCA, different number of eigenvectors were used to create a variable-length features vector. Then, the optimum number of features, corresponding to maximum achieved accuracy, was found. Figure 5.19 plots the recognition accuracy vs. the length of features vector, created by PCA technique. In general, accuracy peaks at a certain interval of features and then diminishes, or saturates. This graph is concerned to Farsi digits recognition (Hoda dataset). In this experiment, the highest accuracies were achieved at intervals of 30 to 60 features.

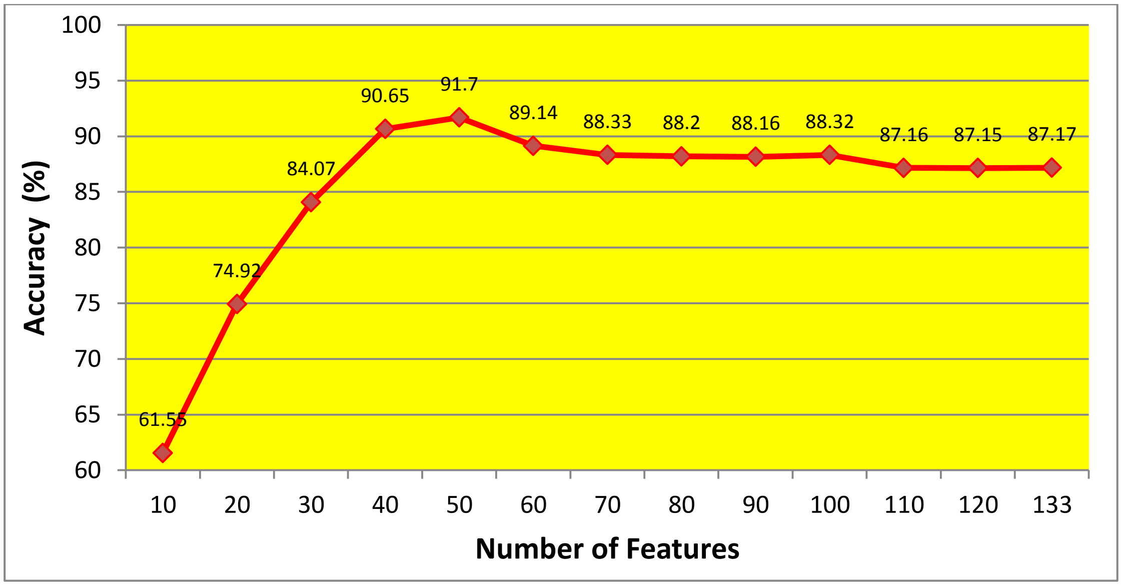


Figure 5.19 : Accuracy vs. the number of features proposed by PCA technique – Farsi digit recognition – Hoda dataset

To find the optimum number of features, the experiment was repeated with different numbers of features at intervals of 30 to 60 features using increment value ‘1’. Finally, the highest accuracy of 93.41% was achieved by using the first 43 features of the features vector. This accuracy was again lower than accuracy 95.12%, achieved by proposed dimensionality reduction method 2S_SA.

5.4 Results Comparison with the Most Related Works, from an OCR View of Point

5.4.1 Farsi Digits

Table 5.6 shows related research works in FOCR domain for recognizing handwritten Farsi digits.

Table 5.6 : Related research works in FOCR domain, digits part

References	Training Dataset, (# of Samples)	# of testing samples	# of features	Classifier	Accuracy %
Mowlaei et al. (2002)	(Private) [only 8 digits]	1600	64	SVM	92.44
Sadri et al. (2003)	CENPARMI version 1	3035	64	MLP-NN	91.25
Mozaffari et al. (2004a)	(Private) [only 8 digits]	1600	64	MLP-NN	91.37
Mozaffari et al. (2005a)	(Private) [only 8 digits]	1600	240	Fr. NN	86.30
Ziaratban et al. (2007)	(Private)	4000	60	MLP-NN	97.65
Ebrahimpor et. al. (2010)	Hoda (6,000)	2,000	81	Mixture of MLPs Expert	91.45
				Mixture of RBFs Expert	95.30
Enayatifar and Alirezanejad (2011)	Hoda (60,000)	20,000	48	MLP-NN	92.70
	Hoda (7,000)	3,000	48	MLP-NN	94.30
PA Method Shayegan and Aghabozorgi (2014)	Hoda (60,000)	20,000	400	*k*-NN	96.49
	Hoda (51,073)	20,000	400	*k*-NN	96.18
	Hoda (36,503)	20,000	400	*k*-NN	95.79
	Hoda (25,726)	20,000	400	*k*-NN	95.26
	Hoda (18,020)	20,000	400	*k*-NN	94.62

The two proposed methods SBR (for dataset size reduction) and 2S_SA (for dimensionality reduction) can be compared to those researches that they used Hoda dataset for training and testing. Hence, the efficiency of the proposed method can be compared with the recent relate researches by Ebrahimpor, Esmkhani and Faridi (2010), Enayatifar and Alirezanejad (2011), and Shayegan and Aghabozorgi (2014). In order to perform more accurate comparisons between the proposed FOCR system and the literature, some new experiments (the fifth block row of Table 5.7) were performed again with similar conditions with the mentioned researches. When the NN classifiers were used, the number of neurons in the input layer was equal to the number of features, and each experiment was repeated 10 times and the obtained results had been reported in average. Table 5.7 is a combination of mentioned related research works of Table 5.6 and new performed experiments by SBR method and 2S_SA method.

Table 5.7 : Related research works in FOCR domain, digits part, Hoda dataset

References		# of Training samples	# of Testing samples	# of features	Classifier	Accuracy %
1	Ebrahimpor et. al. (2010)	6,000	2,000	81	Mixture of MLPs Expert	91.45
2	Enayatifar and Alirezanejad (2011)	60,000	20,000	48	MLP-NN	92.70
		7,000	3,000	48	MLP-NN	94.30
3	PA Method Shayegan and Aghabozorgi (2014)	60,000	20,000	400	*k*-NN	96.49
		51,073	20,000	400	*k*-NN	96.18
		36,503	20,000	400	*k*-NN	95.79
		25,726	20,000	400	*k*-NN	95.26
		18,020	20,000	400	*k*-NN	94.62
4	The Proposed Method SBR	60,000	20,000	400	*k*-NN	96.49
		51,000	20,000	400	*k*-NN	96.33
		36,200	20,000	400	*k*-NN	96.17
		25,200	20,000	400	*k*-NN	95.54
		18,000	20,000	400	*k*-NN	94.91
5	The Proposed Method 2S_SA	60,000	20,000	58	MLP-NN	95.12
		7,000	3,000	58	MLP-NN	96.89
		6,000	2,000	58	MLP-NN	97.34
6	The Proposed Method 2S_SA (10-fold CV)	72,000	8,000	58	MLP-NN	96.93%

Ebrahimpor, Esmkhani and Faridi (2010) employed a mixture of MLPs experts as classifiers in a FOCR system to recognize digits part of the Hoda dataset. They extracted 81 loci features from input images. Also, they only used 6,000 and 2,000 samples of digits part of the Hoda dataset for training and testing their system, and finally, they achieved to 91.45% accuracy. The proposed method 2S_SA was employed in similar conditions with the proposed 58 features for Farsi digits and achieved to higher accuracy (97.34%) compared to the mentioned research (91.45%) (the yellow colored rows in Table 5.7). Here, the higher accuracy was achieved with using lower number of features (58 features against 81 features).

Enayatifar and Alirezanejad (2011) succeeded in recognizing 92.70% of 20,000 testing samples of digit part of the Hoda dataset using a MLP-NN. When they decreased the number of testing samples from 20,000 to 3,000, the accuracy was improved to 94.30% (Table 5.7). The proposed 2S_SA method was employed in similar conditions with using the proposed 58 features for Farsi digits and achieved to higher accuracies (95.12% ; 96.89%) compared to the mentioned research accuracies (92.70% ; 94.30%) in both cases (the green colored rows in Table 5.7). However, here, the number of proposed features, i.e. 58 features, was more than the mentioned research, i.e. 48 features.

Shayegan and Aghabozorgi (2014) used PA method to reduce the volume of training part of the Hoda dataset. Using 400 pixels of each input image, and *k*-NN as classifier engine, they achieved to 96.49% accuracy. The accuracy decreased slightly, when they employed their proposed reduced versions of training dataset (Table 5.7). Here, for all cases of using different number of training samples, the proposed SBR method achieved to higher accuracies (the light brown colored rows in Table 5.7).

At the end, and only for checking the effect of CV technique on accuracy, it was used along with 2S_SA method (the last row of Table 5.7). In this case, the final accuracy increased from 95.12% to 96.93%, showing the positive effect of CV technique on recognition accuracy only in this experiment. However, this result did not compare to the literature, because of doing a fair and accurate comparison process.

5.4.2 Farsi Letters

Table 5.8 summarizes some available literatures in FOCR systems for recognizing handwritten Farsi letters. However, an accurate comparison, from an OCR view of point, between the proposed methodsin thid thesis and literature was not possible, because of using different datasets, different number of training and testing samples, different classifiers, different methods of feature extraction, and different number of extracted features.

The two proposed methods SBR (for dataset size reduction) and 2S_SA (for dimensionality reduction) can be compared to those researches that they used Hoda dataset for training and testing. Hence, the efficiency of the proposed methods can be compared only with the recent related research by Shayegan and Aghabozorgi (2014). Table 5.9 is a combination of related researches in Table 5.8 and the achieved results by SBR method and 2S_SA method.

Table 5.8 : Result comparison for handwritten Farsi letter recognition

Researchers	Training Dataset, (# of Samples)	No. of testing samples	No. of Features	Classifier	Accuracy %
Mowlaei and Faez (2002)	Private, 3,200	2,880	64	MLP-NN	91.81
Mowlaei and Faez (2003)	Private, 3,200	2,880	64	SVM	92.44
Mozaffari et al. (2004a)	Private, 3,200	2,880	64	MLP-NN	87.26
Mozaffari et al. (2004b)	Private, 3,200	2,880	64	SVM	92.00
Mozaffari et al. (2005)	Private, 3,200	2,880	64	SVM	91.33
Shanbehzadeh et al. (2007)	Private, 1,800	1,200	78	Vector Quantization	87.00
Ziaratban et al. (2008)	Private, 11,471	7,647	32, 40, 64, 108	MLP-NN	93.15
Gharoie and Farajpoor (2009)	Private, 125	125	900	MLP-NN	80.00
Jenabzade et al. (2011)	Private, 3,300	3,300	134	MLP-NN Decision Tree	86.30
Rajabi et al. (2012)	IFHCDB, 32,400	16,620	315	ANN,SVM, *k*-NN	97.30
Alaei et al. (2012)	IFHCDB, 36,682	15,338	400	SVM	96.91
PA Method Shayegan and Aghabozorgi (2014)	Hoda 70,645	17,706	400	*k*-NN	80.67
	Hoda 64,755	17,706	400	*k*-NN	80.39
	Hoda 56,613	17,706	400	*k*-NN	78.47
	Hoda 45,978	17,706	400	*k*-NN	72.48
	Hoda 34,293	17,706	400	*k*-NN	64.13

Table 5.9 : Related research works in FOCR domain, characters part, Hoda dataset

References		# of Training samples	# of Testing samples	# of Features	Classifier	Accuracy %
1	PA method	70,645	17,706	400	*k*-NN	80.67
	Shayegan & Aghabozorgi, 2014	64,775	17,706	400	*k*-NN	80.39
		56,613	17,706	400	*k*-NN	78.47
		45,978	17,706	400	*k*-NN	72.48
		34,293	17,706	400	*k*-NN	64.13
2	The Proposed SBR Method	70,645	17,706	400	*k*-NN	80.67
		64,800	17,706	400	*k*-NN	80.46
		56,600	17,706	400	*k*-NN	80.15
		46,000	17,706	400	*k*-NN	79.75
		35,322	17,706	400	*k*-NN	79.36
		34,300	17,706	400	*k*-NN	78.71
		23,548	17,706	400	*k*-NN	74.64
		17,661	17,706	400	*k*-NN	69.46
3	The Proposed 2S_SA Method	70,645	17,706	133	MLP-NN	81.82
		70,645	17,706	93	MLP-NN	83.74
4	The Proposed 2S_SA Method (10-fold CV)	88,300	8,830	93	MLP-NN	86.05

Shayegan and Aghabozorgi (2014) used PA method to reduce the volume of training part of the Hoda dataset. Using 400 pixels of each input image, and *k*-NN as classifier engine, they achieved to 80.67% accuracy. The accuracy decreased, when they employed their proposed reduced versions of training dataset (the first row block of Table 5.9).

In the case of similar number of training samples, the proposed SBR method achieved to better accuracies (the green colored rows in Table 5.9) compared to rival approach PA (the yellow colored rows of Table 5.9). Also, the proposed 2S_SA method achieved to higher accuracy (83.74%), even it used lesser number of features, in comparison to PA method.

At the end, and only for checking the effect of CV technique on accuracy, it was used along with 2S_SA method (the last row of Table 5.9). In this case, the final accuracy increased from 83.74% to 86.05%, showing the positive effect of CV technique on recognition accuracy only in this experiment.

5.5 Error Analysis

An error analysis on misclassified samples showed the main different sources for wrong classification. Some of these error sources are demonstrated as follows:

a) The proposed dataset size reduction SBR technique is based on generating a template for each class. However, there are more than one general shapes for some Farsi letters and digits. For example, Farsi digits 2, 3, 4, 5, and 6 are usually written in two (or more) completely different shapes. For example, 646 out of 6000 (10.77%) training samples of digit ‘4’ of Hoda dataset have been written in pattern ‘٤‘ and the rest have been written in pattern ‘۴‘. As a result, the number of classes for Farsi digits is really 16 instead of 10. Similar to Farsi digits, some Farsi letters, such as ‘س‘ and ‘م‘, are written in more than one shape in handwritten documents (Figure 5.20).

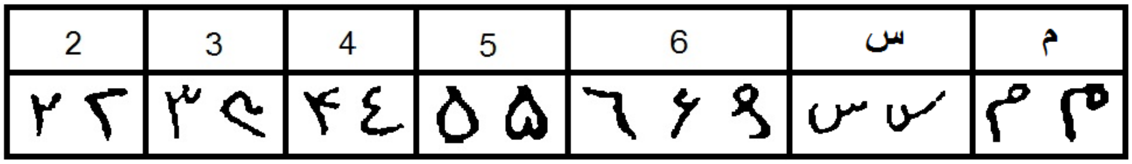


Figure 5.20 : Different shapes for some Farsi digits and letters

These extra patterns degrade the real shapes of generated templates for the aforementioned digits and letters, as some generated templates are not similar enough to the sample images. This characteristic causes the recognition system produces the wrong results. To overcome this drawback, it is necessary to consider more than one template for these patterns.

b) Another major error stems from the degraded samples in the testing part of the Hoda dataset, which cannot even be enhanced during the pre-processing step. Admittedly, there are usually some degraded samples in a standard dataset (taken from real data) that are used to investigate the ability of a PR system to deal with these outlier samples. The Hoda dataset follows this rule. However, some of these samples are too degraded. Figure 5.21 shows the images of some degraded samples of digit ‘4’ that had been misclassified as digit ‘2’ or digit ‘3’. In this case, even an expert cannot recognize them, correctly.

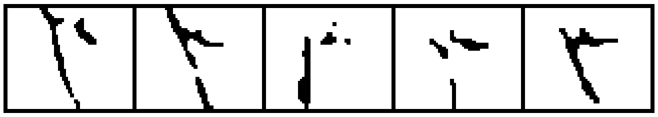


Figure 5.21 : Some degraded samples of digit ‘۴’ (‘4’) which were misclassified as digit ‘۲’ (‘2’) or digit ‘۳’ (‘3’) during recognition process

Table 5.10 and Table 5.11 show a set of degraded samples in the digits and letters parts of the Hoda dataset, in order. These highly degraded samples have a negative impact not only on the template generating process, but also on recognition accuracy.

Table 5.10 : Some degraded digits samples of the Hoda dataset

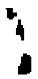
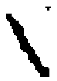
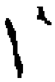
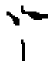
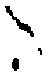
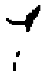
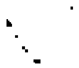
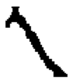
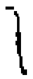
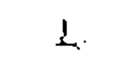
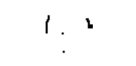
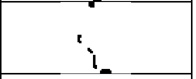
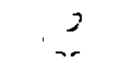
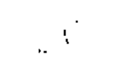
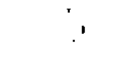
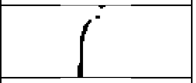
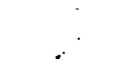
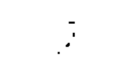
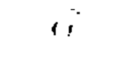
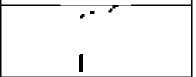
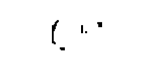
Farsi digits	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
A degraded sample										

Table 5.11 : Some degraded letters samples of the Hoda dataset

Farsi Letters	Degraded Samples		Farsi Letters	Degraded Samples	
ا			ط		
ب			ظ		
پ			ع		
ت			غ		
ث			ف		
ج			ق		
چ			ک		
ح			گ		
خ			ل		
د			م		
ذ			ن		
ر			و		
ز			ه		
ژ			ی		
س			ء		
ش			آ		
ص			ھ		
ض			ـه		

c) Another problem concerning FOCR systems is the excessive similarity between the original shapes of some character group sets, especially in handwritten mode (Table 2.2). Most of the FOCR systems suffer from this too similarity characteristic. For this reason in some automated applications, such as mailing (in Iran), digit '۲' ('2') is not used in zip codes (Mozaffari, Faez & Rashidy Kanan, 2004a). Table 5.12 shows the confusion matrix for one of the performed experiments in Section 4.4.5.1 (the first row of Table 4.2) for Farsi digits recognition. Among all occurred errors (there was no rejection strategy in this research), 55.27% were related to misclassification of similar digits '2', '3', and '4' (the yellow cells), and 15.38% were related to misclassification of similar digits '1', '6', and '9' (the blue cells).

Table 5.12 : Confusion matrix for Farsi digits recognition (Section 4.4.5.1)

	۰ (0)	۱(1)	۲(2)	۳(3)	۴(4)	۵(5)	۶(6)	۷(7)	۸(8)	۹(9)
۰ (0)	2,000	0	0	0	0	0	0	0	0	0
۱ (1)	0	1,995	5	0	0	0	0	0	0	0
۲ (2)	0	27	1,899	46	15	2	2	9	0	0
۳ (3)	0	0	143	1,801	47	2	0	4	3	0
۴ (4)	3	8	71	66	1,834	4	7	3	0	4
۵ (5)	0	7	0	0	3	1,966	3	2	14	5
۶ (6)	2	16	10	2	3	0	1,932	5	2	28
۷ (7)	0	3	12	2	3	0	4	1,974	0	2
۸ (8)	0	7	0	0	0	0	6	0	1,980	7
۹ (9)	0	28	8	0	0	5	36	0	6	1,917

d) Dots and extensions of handwritten letters are often not located at the exact position on top, middle or under the main part of the letters. Hence, some important features, related to dots and extensions of letters, are changed dramatically, and as a result, these features cannot play an effective role in recognition process.

5.6 Summary

This chapter compared the proposed techniques SBR for dataset size reduction and 2S_SA for dimensionality reduction to recent researches in the mentioned domain, in a FOCR application. In order to carry out accurate comparisons, the experiments were performed in maximum similarity with the conditions in the literatures. However, sometimes it was not possible to create similar conditions. The results from OCR view of points showed the digits recognition achieved to higher accuracy compared to letters recognition. Some important reasons for this result will be discussed in Chapter 6.

Although, the PA method for dataset size reduction (Section 2.5.1, Appendix VI) succeeded to decrease processing time (speeding up the system) without a significance loss in system accuracy, but the proposed dataset size reduction SBR outperformed the PA method. The main reason for this improvement is the proposed sieving method not only keeps some samples near to classes centers (which they are necessary to make system model), but also keeps some boundaries samples in the final reduced dataset. The existence more samples near to class centers help to make more accurate system model. Also, sampling operation causes to create more distance between each two successive samples in the final training dataset. These two characteristics affect on final accuracy, positively. In other word, the SBR method guaranties to make a smaller size of dataset for the best accuracy, because by sampling operation, based on the similarity values concept, a new version of dataset with more distribution of samples is made, and a recognition engine (such as *k*-NN) with more spread entries outperforms than a recognition engine with closer entries.

2S_SA dimensionality reduction method achieved to higher accuracy compared to rival techniques. Although, this technique is time consuming process, but the system training is carried one time and the system testing is carried several times. However, it is taken in two stages, to decrease the overall processing time.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Introduction

As discussed in Section 2.5, the large volume of training datasets and high dimensionality of features space causes time complexity in both the training and testing phases in different PR systems. Hence, the main focus of this research was reducing the volume of datasets (in order to increase the recognition speed), and reducing the dimensionality of features vector (in order to increase the system speed and accuracy) in a FOCR system framework.

For dataset size reduction, the new method SBR was proposed. In order to increase recognition accuracy, a small features vector was found using the new dimensionality reduction technique 2S_SA. The superiority of the proposed dimensionality reduction technique 2S_SA was shown with comparison the achieved results with the obtained results by other dimensionality reduction methods such as PCA and RP techniques.

To evaluate the performance of the proposed methods SBR and 2S_SA, some experiments were conducted on the benchmark datasets Hoda and MNIST. All techniques were designed, implemented, tested, and compared to the literature to perform the task of character recognition. The summary of finding, limitation of the research, and recommendation are given in the following sub-sections.

6.2 Summary of Results and Findings

First, the reasons for low speed and low accuracy of FOCR systems (main problems) are stated based on all the findings in the literature review (Chapter 2) and the evaluation chapter (Chapter 5). The reasons are drawn by answering the following questions corresponding to main problems statement:

* What is the reason for low accuracy of a FOCR system?

1) The existence of some deficiencies in output of pre-processing block decreases the recognition accuracy (corresponding to sub-problem 1).

2) Large number of less-important features (attributes) for training and testing samples, which they decrease the final recognition accuracy (corresponding to sub-problem 3).

In Section 4.3.6, the effect of image enhancement, by applying necessary pre-processing operations, was described. It was shown that the proposed pre-processing operation CBP, for connecting the broken parts of an image, had positive effect in final accuracy (Table 4.1). After applying CBP method, the recognition accuracy was improved 6.63% for Farsi digits, and 5.85% for English digits. The achieved result of applying the proposed CBP method on input data was compared to traditional pre-processing approach (Table 4.1).

In Section 4.6.2, it was shown the proposed dimensionality reduction method 2S_SA has positive effect in both recognition time and system accuracy by explaining the different steps of the method. Here, based on the literature, a set of the most-used features, in OCR domain, was extracted in order to simulate similar conditions with the literature. The proposed method 2S_SA found a small effective features vector from the large initial features set. As a result, the accuracies were increased 4.71%, 1.92%, and 2.95% for recognition of Farsi digits, Farsi letters, and English digits, respectively. The overall system time was also decreased correspondingly to decrease of number of features in features space. The achieved results by 2S_SA method were compared with other traditional dimensionality reduction methods PCA and RP techniques (Table 5.5). Related experiments in Chapter 5 showed that 2S_SA method outperformed the other rival methods. The main reason that 2S_SA method outperforms other rival techniques is, it chooses the best small subset of initial features vector, by investigating the existing correlation between them.

* <u>What is the reason for low speed of a FOCR system?</u>

1) Large number of less-important training samples in datasets, which they increase the training time and also testing time, especially when a specific classifier such as *k*-NN is used (corresponding to sub-problem 2).

2) High dimensionality of features space (the existence of some less-important features in features vectors), which they increase the training and testing time, especially when using a specific recognition engines such as *k*-NN (corresponding to sub-problem 3).

In Section 4.4.5, the effect of dataset size reduction operations on system accuracy and recognition speed was described. The proposed size reduction method SBR was applied on Hoda (Section 4.4.5.1, Section 4.4.5.2), and then, the recognition accuracy was measured. For recognizing Farsi digits, the training dataset size was decreased to about half (recognition speed was increased to about double), whereas the accuracy decreased only 0.68% (Figure 4.16). For recognizing Farsi letters, the training dataset size was decreased to about half, but the accuracy decreased 1.31% (Figure 4.17). The achieved results were compared to most-related works in the literature (Figure 5.11, Figure 5.17).

In summary, the best achieved results in this research were:

- Enhancing 6.63% in final accuracy for Farsi digits recognition, and 5.85% in final accuracy for English digits recognition, by applying CBP method in pre-processing step (Section 4.3.6).
- Reduction the dataset size to 50% of initial volume for Farsi digits, meanwhile the accuracy was decreased only 0.68% (Section 4.4.5.1), which this result was better than similar research work in this domain (Figure 5.11).
- Reduction of dimensionality of features space for Farsi digits and letters, and for English digits, and as a result, increasing the accuracy in all cases (Table 4.6, Figure 4.29, Figure 4.30, Figure 4.31)

6.3 Achievement of the Objectives

The following are the objective of this research:

1) To enhance the output quality of pre-processing block. In order to fulfill this objective, the following methods were developed:

- Developing a method to estimate pen width for handwritten digits and letters (Equation 4.1).
- Developing a method to connect the broken parts of an image together (CBP method), in order to increase the output quality of the pre-processing block. (Section 4.3.3, Algorithm 1).

The mentioned subjects were explained in Section 4.3.3.

2) To propose a new technique for dataset size reduction to speed up system training and testing. In order to fulfill this objective, the following methods were developed:

- Developing a method to generate a template for each class of training samples (Section 4.4.1, and Equation 4.3).
- Developing a distance measurement function for template matching purpose (Equation 4.5).
- Developing a method for reducing the number of training samples (SBR method) (Section 4.4.4, Algorithm 2).

The mentioned subjects were explained in Section 4.4.

3) To propose a new technique for dimensionality reduction to speed up system training and testing, and increasing the system accuracy. In order to fulfill this objective, the following methods were developed:

- Developing 1D_SD and 1D_MM tools to analyze the one-dimensional spectrums diagram for each feature in features space (Section 4.6.1.1).
- Developing 2D_SD and 2D_MM tools to analyze the two-dimensional spectrums diagrams for each couple of features in features space (Section 4.6.1.2).
- Developing a two-stage algorithm for reducing the number of features (2S_SA method) (Section 4.6, Algorithm 3, Algorithm 4).

The mentioned subjects were explained in Section 4.6.

4) To test and evaluate the capability of the proposed methods in improving the performance of FOCR systems, by applying it on Farsi digits and letters. In order to fulfill this objective, the proposed model was validated by using two benchmark datasets Hoda and MNIST, and the results were evaluated by comparing them to some key-researches in the domain (Chapter 4, Chapter 5).

6.4 Contributions of this Research

By focusing on the available methods for FOCR systems, the problems were identified: weakness in output of pre-processing step, large volume of datasets with a large number of less-important samples, and high dimensionality of features space with less-important features. Hence, some extra steps were added to general model of OCR systems (Figure 3.4) to create a new model for a FOCR system (Figure 3.5). Accordingly, the major contributions of this thesis research are outlined as follows:

- #1: Proposing a new model for FOCR systems (Figure 3-5).
- #2: Finding a new approach to estimate pen width of handwritten Farsi digits and letters (Equation 4.1).
- #3: Proposing the new method CBP (Algorithm 1; Section 4.3.3) to connect the broken parts of an image together, in order to enhance the quality of output of pre-processing stage.
- #4: Proposing the new similarity measurement function Equation 4.5, to compute similarity between a sample and corresponding class template.
- #5: Proposing the new method SBR (Figure 3.2; Section 4.4) for sieving the training part of a dataset, in order to speed up the overall processing time.
- #6: Proposing the new method 2S_SA (Figure 3.3; Section 4.6) to reduce the dimensionality of features vector, in order to speeding up the recognition process and to increase the system accuracy.
- #7: Finding a small feature set suitable for FOCR systems, in order to increase system accuracy (Appendix VII).

<u>A new model for FOCR systems</u>: The main goals of this study were decreasing the training size of a dataset, and increasing the recognition accuracy of a FOCR system. To such an aim, a framework was suggested for handwritten Farsi letters and digits recognition. In the proposed model, multiple steps were carried as follows:

- Pre-processing operations, because it increases the quality of input samples, that it will caused in improvement system accuracy.
- Dataset size reduction, because it decreases the overall processing time (it increases the system speed).
- Feature extraction operation, because dealing with features are usually produced the better results compared to using the input images, directly.
- Dimensionality reduction, because it decreases the overall processing time (it increases the system speed), and increasing the system accuracy with removing non-suitable features.
- Recognition process, to validate the proposed techniques.

A new approach to estimate pen width: In the pre-processing step, it is needed to connect the broken parts of an image together (image enhancement). The connector lines should have the same width of the other parts in the image. Hence it is necessary to find the pen width of writing.

A new method for connecting the broken parts of an image together (CBP method): The proposed CBP method estimates the pen width in three different ways (Equation 4.1). Then, by utilizing the connected component analysis, it traverses the outer contour of the separated blocks in an image and connects them together.

A new similarity measurement function: For dataset size reduction task, it is necessary to rearrange the training samples based on their similarity values to corresponding class templates. To such an aim, by using modified frequency diagram matching, a new similarity measurement function (Equation 4.5) is used to compute similarity between a sample and its class template.

A new method for sieving the training part of a dataset (SBR method): One of the main objectives in this research was decreasing the size of training datasets. The proposed sieving method reduced the re-arranged training dataset samples in different sampling rates.

A new two-stage method to reduce dimensionality of features space (2S_SA method): Another main objective of this thesis was increasing the recognition accuracy and recognition speed, by removing some less-important features from features space. It was carried out by applying the proposed dimensionality reduction method 2S_SA on features space, by using tools 1D_SD, 1D_MM, 2D_SD, and 2D_MM.

Finding a small features set suitable for FOCR systems: The output of the proposed reduction method 2S_SA was a small features set, suitable to use in FOCR systems.

6.5 Limitation of the Current Study

1) This research only managed to focus on free-access handwritten Farsi Hoda dataset. However, for the sake of comparison and evaluation, the handwritten English digits dataset MNIST was employed, too.

2) Here, the alone mode of Farsi letters was only used, because to handle Farsi words, the internal segmentation operation is needed, and segmentation operation was out of this research scope.

3) Although, it was shown the achieved results in this research were better than the similar researches in the related domain, but the results are still far from final goals in OCR

systems, i.e. the accuracy 99.9% with recognition of at least 5 characters per second (Khorsheed, 2002).

4) The proposed dimensionality reduction method 2S_SA is a time consuming process, because it needs to create 1D_SD and 1D_MM spectrums diagrams for all individual features, and 2D_SD and 2D_MM spectrums diagrams for all feature pairs of the initial extracted features. However, the training process is performed only one time, and the trained system is used several times.

6.6 Conclusion

Based on the literature review, research background, and the obtained results from various experiments presented throughout this thesis, it can be concluded that this research work has achieved to its objectives as described in Section 6.3. The performed experiments in each section were carried out regardless the previous sections' output, in order to find the effect of each proposed method, separately. However, the output of the pre-processing step (enhanced input images) was fed to all next steps, i.e. dataset size reduction, feature extraction, dimensionality reduction, and recognition.

Misrecognizing is not a problem unique to computers; even human beings have difficulty in recognizing some scripts and have an error rate 4% on reading tasks in the absence of context (Zeki, 2005). These errors are mainly the result of variation in shapes related to the writing habits, styles, education, social environment, health, psychological situation and other conditions affecting the writer; as well as other factors such as writing instrument, writing surface, scanning algorithms and machines.

At the end, it can be claimed that this research has contributed to the available corpus of knowledge in PR domain, particularly of handwritten FOCR systems. This has been achieved by proposing a new method for estimating pen width, the new technique CBP for connecting the broken parts of an image (enhancement of the output of pre-processing step; objective #1), proposing the new dataset size reduction method SBR, in order to decrease the training volume size and increase the recognition speed (objective #2), and proposing the new method 2S_SA to decrease dimensionality of a numerical features vector, in order to increase the system accuracy and speeding up the recognition system (objective #3).

The proposed method for pen width estimation is regardless of Farsi characters, and it can be extended to other languages OCRs, such as Chinese characters.

The salient point of the proposed dataset size reduction approach SBR is that it is not only effective in OCR application, as a subcategory of PR systems, but also it can be used with appropriate adjustments, for other PR systems with different types of pictorial datasets.

The proposed dimensionality reduction method 2S_SA shows the correlation between each feature pairs, and it can identify which features are selected for final features vector. It performed very well for FOCR systems, and it not only decreased the features space dimensions, but also increased the final accuracy. In addition, the achieved results showed the superiority of the proposed features set by 2S_SA method, against other features set produced by other conventional features selection methods in similar conditions. The reason of this superiority is 2S_SA method looks for the best features (non-overlap features) in a features vector by analyzing the correlation between the features. Although the results have been reported for OCR application, the salient point of the proposed 2S_SA method is that it can also be used for other datasets for each domain, with numerical features vectors. Table 6.1 depicts this research activity in a very compact scheme. Also, Table 6.2 shows the best achieved results in this study, in a comparison way and in a condensed mode representation.

Table 6.1 : A brief review on this research, problems statements, objectives, the proposed methods, and contributions.

Main Problems	Sub-problems	Objectives	Proposed Methods	Contributions
Low System accuracy	Weakness of pre-processing operations	#1 Enhancing the quality of pre-processing output	CBP	#2 :Pen Width Estimation #3: Proposing CBP Method
	High Dimensionality	#3 Dimensionality Reduction	2S_SA	#6: Proposing 2S_SA method #7: Proposing a small features set for FOCR systems
Low System Speed	Large Training Datasets	#2 Dataset Size Reduction	SBR	#4: Proposing a new similarity measurement function #5: Proposing SBR method
	High Dimensionality	#3 Dimensionality Reduction	2S_SA	
		#4 System Test and Evaluation	All the methods	#1: Proposed a new model for FOCR system

Table 6.2 : A brief review on the achieved results in this thesis

		Pre-processing	Size Reduction (to 50% of original dataset)	Dimensionality Reduction (58 Features)
Accuracy	Related Works	Traditional : noise removal, slant correction, scaling, translation : 83.77%	Vishwanathan : 96.82%	PCA : 89.00%
			Cervantes : 98.88%	
			PA : 99.00%	RP : 83.66%
	The Proposed Methods	Traditional + CBP method : 90.40%	SBR : 99.30%	2S_SA : 95.12%

6.7 Future Works

There are some researches efforts have been conducted for developing FOCR systems, but there is a long way to achieve ultimate goals. Future researches in FOCR domain are being set out by this thesis can be considered in the following directions:

- Utilizing more handwritten Farsi datasets to investigate the effects of the proposed size reduction technique on a set of datasets.
- Extending the proposed connecting broken parts of an image, for Farsi letters. It will help to achieve higher accuracy in letter recognition part.
- Extending the proposed dimensionality reduction technique from two-dimensional state to n-dimensional (n: total number of initial extracted features).
- The achieved accuracy by using one classifier is usually acceptable, but additional classifiers can help to correct some errors, and also recognizing of rejected samples. Hence, ensemble classification technique should be considered in FOCR domain, too.
- By analyzing the confusion matrix (Table 5.12) regarding to Farsi digits recognition experiments, it was understood that the majority of occurred errors were produced by some digits groups with similar body, such as ‘۲‘, ‘۳‘, and ‘۴‘. The same situation was seen for Farsi letters, too. Hence, merging the similar digits in a class, and using a multi-stage recognition system can be considered as a good goal.
- In spite of the intense research effort made on character recognition, and the success achieved in specific aspects of OCR systems, no ideal solution for Farsi cursive script segmentation problem has yet been found, and the current available results in this part are very far from the final goal (Appendix II). Some algorithms that have been developed for words segmentation in other cursive languages can be applied to

Farsi handwriting, but these algorithms are generally not fully suitable for handling the Farsi manuscripts. Moreover, the performance of segmentation blocks in FOCR systems is lower than the performance in Latin OCR systems. The special characteristics of the Farsi language have been an impediment in all analytical FOCR systems. It is clear that using holistic recognition methods to recognize handwritten documents in cursive languages are suitable for handling limited lexicons, but these methods usually do not produce acceptable results when they try to handle very big vocabulary sets. Hence, the current segmentation algorithms need to be further improved and further researches on this aspect are necessary to resolve this important problem.

REFERENCES

Abandah, G.A., & Anssari, N. (2009). Novel Moment Features Extraction for Recognizing Handwritten Arabic Letters. *Journal of Computer Science*, *5*(3), 226-232.

Abandah, G. A., Jamour, F. T., & Qaralleh, E. A. (2014). Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks. *International Journal on Document Analysis and Recognition*, 1-17.

Abandah, G.A., Younis, K.S., & Khedher, M.Z. (2008). Handwritten Arabic character recognition using multiple classifiers based on letter form. In *Proceeding of 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*, pp. 13-16.

Abdul Sattar, S., & Shahl, S. (2012). Character Recognition of Arabic Script Languages. *The Second International Conference on Communication and Information Technology*, pp. 502-506.

Abdullah, M.A., Al-Harighi, L.M., & Al-Fraidi, H.H. (2012). Offline Arabic Handwriting Character Recognition Using Word Segmentation. *Journal of Computing*, *4*(*3*), 40-44.

Abedi, A., Faez, K., & Mozaffari, S. (2009). Detecting and Recognizing Numerical Strings in Farsi Document Images. *24th IEEE International Conference of Image and Vision Computing*, pp. 403-408.

Abuhaiba, I.S.I. (2006). Efficient OCR using simple features and decision trees with backtracking. *The Arabian Journal for Science and Engineering*, *31*(2B), 223-243.

Achlioptas, D. (2001). Database-friendly random projections. Symposium on Principles of Database Systems, pp. 274-281.

Al-A'ali, M., & Ahmad, J. (2007). Optical Character Recognition System for Arabic Text Using Cursive Multi-Directional Approach. *Journal of Computer Science*, *3*(7), 549–555.

Al-Hajj, R., Likforman, L., & Mokbel, C. (2009). Combining Slanted Frame Classifiers for Improved HMM_Based Arabic Handwriting Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *31*(7), 1165-1177.

Al-Khateeb, J.H., Jiang, J., Ren, J., Khelifi, F., & Ipson, S.S. (2009). Multiclass Classification of Unconstrained Handwritten Arabic Words Using Machine Learning Approaches. *The Open Signal Processing Journal*, *2*, 21-28.

Al-Khateeb, J.H. (2012). Offline Handwritten Arabic Digit Recognition Using Dynamic

Bayesian Network. *The 1st International Conference on Computing and Information Technology (ICCIT 2012)*, pp. 176-180.

Al-Shatnawi, A., & Omar, Kh. (2008). Methods of Arabic Language Baseline Detection – The State of Art. *International Journal of Computer Science and Network Security, 8*(10), 137-143.

Al-Shatnawi, A., & Omar, KH. (2009). A Comparative Study between Methods of Arabic baseline Detection. *IEEE International Conference on Electrical Engineering and Informatics, ICEEI'09*, pp. 73-77.

Al-Tameemi, A.M., Zheng, L., & Khalifa, M. (2011). Offline Arabic Words Classification using Multi Set Features. *Information Technology Journal*, 1-7.

Alaei, A., Nagabhushan, P., & Pal, U. (2009a). Fine Classification of Unconstrained Handwritten Persian/Arabic Numerals by Removing Confusion amongst Similar Classes. *10th IEEE International Conference on Document Analysis and recognition*, pp. 601-605.

Alaei, A., Pal, U., & Nagabhushan, P. (2009b). Using Modified Contour Features and SVM Based Classifier for the Recognition of Persian/Arabic handwritten Numerals. *7th IEEE International Conference on Advanced in Pattern recognition,* pp. 391-394.

Alaei, A., Nagabhushan, P., & Pal, U. (2010a). A New Two Stage Scheme for the Recognition of Persian Handwritten Characters. *12th IEEE International Conference on Frontiers in Handwriting Recognition,* pp. 130-135.

Alaei, A., Nagabhushan, P., & Pal, U. (2010b). A Baseline Dependent Approach for Persian Handwritten Character Segmentation. *20th IEEE International Conference on Pattern Recognition,* pp. 1977-1980.

Alaei, A., Nagabhushan, P., & Pal, U. (2011a). A New Dataset of Persian Handwritten Documents and its Segmentation. *7th IEEE Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 1-5.

Alaei, A., Nagabhushan, P., & Pal, U. (2011b). Piece-wise painting technique for line segmentation of unconstrained handwritten text: a specific study with Persian text documents. *Pattern Analysis and Applications, 14,* 381-394.

Alaei, A., Pal, U., & Nagabhushan, P. (2011). A new scheme for unconstrained handwritten text line segmentation. *Pattern Recognition, 44*(4), 917-928.

Alaei, A., Pal, U., Nagabhushan, P., & Kimura, F. (2011). A Painting Based Technique for Skew Estimation of Scanned Documents, *IEEE International Conference on Document Analysis and Recognition*, pp. 299-303.

Alaei, A., Pal, U., & Nagabhushan, P. (2012). A Comparative Study of Persian/Arabic Handwritten Character Recognition. *IEEE International Conference on Frontiers in Handwriting Recognition,* pp. 123-128.

Alavipour, F., & Broumandnia, A. (2014). Farsi Character Recognition using New Hybrid Feature Extraction Methods. *International Journal of Computer Science, Engineering & Information Technology*, *4*(1), 15-25.

Alginahi, Y.M., & Siddiqi, A. (2010). Multi stage hybrid Arabic/Indian numeral OCR system. *International Journal of Computer Science and Information Security, 8*(1), 9-18.

Alginahi, Y.M. (2012). A Survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition,* DOI 10.1007/s10032-012-0188-6.

Alirezaee, Sh., Aghaeinia, H., Ahmadi, M., & Faez, K. (2004a). An efficient selected feature set for the middle age Persian character recognition. *33rd International Symposium on Applied Imagery Pattern Recognition Workshop*, pp. 246-250.

Alirezaee, Sh., Aghaeinia, H., Ahmadi, M., & Faez, K. (2004b). Recognition of middle age Persian characters using a set of invariant moments. *33rd International Symposium on Applied Imagery Pattern Recognition Workshop*, pp. 196-201.

Alirezaee, Sh., Aghaeinia, H., Faez, K., & Rashidzadeh, R. (2005). An efficient preprocessing block for the middle-age Persian manuscripts. *IEEE Canadian Conference on Electrical and Computer Engineering, 1557418*, pp. 2170-2173.

Arica, N., & Yarman, F.T. (2002). Optical character recognition for cursive handwriting. *IEEE Pattern Analysis and Machine Intelligence, 24*(6), 801-813.

Ashkan, M.Y., Guru, D.S., & Punitha, P. (2006). Skew estimation in Persian documents: A novel approach. *Proceedings of IEEE Conference Computer Graphics, Imaging and Visualization: Techniques and Applications, 1663769,* July, pp. 64-70.

Azizi, N., Farah, N., Khadir, M., & Sellami, M. (2009). Arabic handwritten word recognition using classifiers selection and features extraction/selection. *17th IEEE Conference in Intelligent Information System*, pp. 735-742.

Azmi, R., & Kabir, E. (2001). A new segmentation technique for ominifont Farsi text. *Pattern Recognition Letters*, *22* (2), 97-104.

Azmi, R., Pishgoo, B., Norozi, N. Koohzadi, M., & Baesi, F. (2010). A hybrid GA and SA algorithms for feature selection in recognition of handprinted Farsi characters. *IEEE International Conference on Intelligent Computing and Intelligent Systems, 3*, pp. 384-387.

Baghshah, M.S., M., Shouraki, S.B., & Kasaei, S. (2005). A novel fuzzy approach to recognition of online Persian handwriting. *5th IEEE International Conference on Intelligent Systems Design and Applications, 1578796*, pp. 268-273.

Baghshah, M. S., Shouraki, S. B., & Kasaei, S. (2006). A novel fuzzy classifier using fuzzy LVQ to recognize online persian handwriting. *2nd IEEE International Conference on Information and Communication Technologies, 1*, pp. 1878-1883.

Bahmani, Z., Alamdar, F., Azmi, R., & Haratizadeh, S. (2010). Offline Arabic/Farsi Handwritten Word Recognition Using RBF Neural Network and Genetic algorithm. *IEEE International Conference on Intelligent Computing and Intelligent Systems, 3*, pp. 352-357.

Bansal, A., Mehta, K., & Arora, S. (2012). Face Recognition using PCA & LDA Algorithms. *2nd* IEEE *International Conference on Advanced Computing & Communication Technologies,* pp. 251-254.

Basu, B., Chaudhuri, C., Kundu, M., Nasipuri, M., & Basu, D.K. (2007). Text line extraction from multi-skewed handwritten documents. *Pattern Recognition, 40*, 1825-1839.

Benmokhtar, R., Delhumeau, J., & Gosselin, P.H. (2013). Efficient Supervised Dimensionality Reduction for Image Categorization. *IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 2425-2428.

Bidgoli, A.M., & Sarhadi, M. (2008). IAUT/PHCN: Azad University of Tehran/Persian Handwriting City Names, a very large database of Handwritten Persian Word. *11th International Conference on Frontiers in Handwriting Recognition,* pp. 192-197.

Bouchareb, F., Hamdi, R., & Bedda, M. (2008). Handwritten Arabic character recognition based on SVM classifier. *3rd* IEEE *International Conference on Information and communication Technologies: From Theory to Application,* pp. 1-4.

Boucheham, B. (2012). PLA – Data Reduction for Speeding up Time Series Comparison. *International Arab Journal of Information Technology, 9*(5), 117-121.

Bronoski Borges, H., & Nievola, J.C. (2012). Comparing the dimensionality reduction methods in gene expression databases. *Expert Systems with Applications*, 39, 10780-10795.

Broumandnia, A., & Shanbehzadeh, J. (2007). Fast Zernike wavelet moments for Farsi character recognition. *Image and Vision Computing, 25*(5), 717-726.

Broumandnia, A., Shanbehzadeh, J., & Nourani, M. (2007). Segmentation of printed Farsi/Arabic Word. *IEEE International Conference on Computer systems and Applications*, pp. 761-766.

Bunke, H., & Wang, P.S.P. (1997). Handbook of Character Recognition and Document Image Analysis. *World Scientific Publishing Co. Pte. Ltd.*, Singapore.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 1-43.

Cano, J.R., Garcia, S., & Herrera, F. (2008). Subgroup discover in large size data sets preprocessed using stratified instance selection for increasing the presence of minority classes. *Pattern Recognition Letters*, *29*(2008), 2156-2164.

Cervantes, J., Li, X., & Yu. W. (2008). Support Vector Classification for Large Data Sets by Reducing Training Data with Change of Classes. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics,* pp. 2609–2614.

Chavan, M.R.S., & Sablei, G.S. (2013). An Overview of Speech Recognition Using HMM. *International Journal of Computer Science and Mobile Computing*. *2*(6), 233-238.

Curic, V., Lindblad, J., Sladoje, N., Sarve, H., & Borgefors, G. (2012). A new set distance and its application to shape registration. *Pattern Analysis and Applications, 12*, 1-12.

Dasgupta, S., & Gupta, A. (1999). An elementary proofs of the Johnson-Lindenstrauss lemma. Technical report TR-99-006, International Computer Science Institute, Berkeley, California, USA.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(3), 131-156.

De Stefano, C., Fontanella, F., Marrocco, C., & Scotto di Freca, A. (2014). A GA-based feature selection approach with an application to handwritten character recognition. *Pattern Recognition Letters*, *35*, 130-141.

Deegalla, S., & Bostrom, H. (2006). Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification. *5th IEEE International Conference on Machine Learning and Applications,* pp. 245-250.

Deepu, V., Sriganesh, M., & Ramakrishnan, A.G., (2004). Principal Component Analysis for Online Handwritten Character Recognitions. *17th International Conference on Pattern Recognition, 2,* pp. 327-330.

Dehbovid, H., Razzazi, F., & Alirezaee, Sh. (2010). A Novel Method for De-warping in Persian Document Images Captured by Cameras. *International Journal of Image Processing, 4*(4), 390-400.

Dehghan, M., Faez, K., Ahmadi, M., & Shridhar, M. (2000). Off-line unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov word models. *15th IEEE International Conference on Pattern Recognition, 2,* pp. 351-354.

Dehghan, M., Faez, K., Ahmadi, M., & Shidhar, M. (2001a). Unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov models. *Pattern Recognition Letters, 22* (2), 209-214.

Dehghan, M., Faez, K., Ahmadi, M., & Shidhar, M. (2001b). Handwritten Farsi (Arabic) word recognition: A holistic approach using discrete HMM. *Pattern Recognition, 34*(5), 1057-1065.

Dehghani, A., Shabani, F ., & Nava, D. (2001). Offline Recognition of Isolated Persian Handwritten C haracters using Multiple Hidden Markov Model. *IEEE International Conference on Information Technology: Coding and Computing,* pp. 506-510.

Dhandra, B.V., Malemath, V.S., Mallikarjun, H., & Hegadi, R. (2006). Skew Detection in Binary Image Documents Based on Image Dilation and Region Labeling Approach. *18th IEEE International Conference on Pattern Recognition,* pp. 954-957.

Ding, Ch., & He, X. (2004). K-means Clustering via Principal Component Analysis. *21th International Conference on Machine Learning,* pp. 1-9.

Dinges, L., Al-Hamadi, A., Elzobi, M., Al-Aghbari, Z., & Mustafa, H. (2011). Offline Automatic Segmentation based Recognition of Handwritten Arabic Words. *International Journal of Signal Processing, Image Processing, and Pattern recognition, 4*(4), 131-143.

Downton, A.C., Kabir, E., & Guillevic, D. (1988). Syntactic and contextual post processing of handwritten addresses for optical character recognition. *9th IEEE International Conference on Pattern Recognition*, pp. 1072-1076.

Ebrahimi, A., & Kabir, E. (2008). A pictorial dictionary for printed Farsi sub words. *Pattern Recognition Letters*, *29*, 656-663.

Ebrahimpor, R., Esmkhani, A., & Faridi, S. (2010). Farsi Handwritten digit recognition based on mixture of RBF experts, *IEICE Electronics Express*, *7*(14), 1014-1019.

Ehsani, M.S., & Babaee, M.R. (2006). Recognition of Farsi Handwritten Cheque Values using Neural Networks. 3rd *IEEE International Conference on Intelligent Systems*, pp. 656-660.

El-Abed, H., & Matgner, V. (2007). Comparison of Different Preprocessing and Feature Extraction Method for Offline Recognition of Handwritten Arabic Words. *9th IEEE International Conference on Document Analysis and Recognition, 2,* pp. 974-978.

Elbaati, A., Kherallah, M., Ennaji, A., & Alimi, A.M. (2009). Temporal Order Recovery of the Scanned Handwriting. *10th IEEE International Conference on Document Analysis and Recognition*, pp. 1116-1120.

El-Hajj, R., Likforman, L., & Mokbel, C. (2005). Arabic Handwriting Recognition Using Baseline Dependent Features and Hidden Markov Modeling. *8th IEEE International Conference on Document Analysis and Recognition, 2,* pp. 893-897.

El-Glaly, Y., & Quek, F. (2011). Isolated Handwritten Arabic Character Recognition using Multilayer Perceptrons and K Nearest Neighbor Classifiers. pp. 1-6.

Elnagar, A., & Bentrica, R. (2012). A Multi Agent Approach to Arabic Handwritten Text Segmentation. *Journal of Intelligent Learning Systems and Application, 4*, pp. 207-215.

Elzobi, M., Al-Kamdi, A., Dinges, L., & Michaelis, B. (2010). A Structural Features Based Segmentation for Offline Handwritten Arabic Text. *5th IEEE International Symposium on Image/Vision Communication over Fixed and Mobile Networks*, pp. 1-4.

Enayatifar, R., & Alirezanejad, M. (2011). Offline Handwriting Digit Recognition by using Direction and Accumulation of Pixels. *International Conference on Computer and Software Modeling, 14*, pp. 214-220.

Fan, K.C., Chen, D.F., & Wen, M.G. (1998). Skeletonization of binary images with nonuniform width via block decomposition and contour vector matching. *Pattern Recognition, 1*(7), 823-838.

Faradji, F., Faez, K., & Nosrati, M.S. (2007). Online Farsi Handwritten Words Recognition Using a Combination of 3 Cascaded RBF Neural Networks. *IEEE International Conference on Intelligent and Advanced Systems*, pp. 134-138.

Faradji, F., Faez, K., & Mousavi, M.H. (2007). An HMM-based Online Recognition System for Farsi Handwritten Words. *IEEE International Conference on Intelligent and Advanced Systems*, pp. 1187-1192.

Farrahi Moghaddam, R., Cheriet, M., Adankon, M. M., Filonenko, K., & Wisnovsky, R. (2010). IBN SINA: a database for research on processing and understanding of Arabic manuscripts images. *9th International Workshop on Document Analysis Systems*, pp. 11-18.

Farouq, F., Gouindaraju, V., & Perrone, M. (2005). Pre–processing Methods for Handwritten Arabic Documents. *8th IEEE International Conference on Document Analysis and Recognition*, pp. 267-271.

Feng, B. Y., Ren, M., Zhang, X. Y., & Suen, C. Y. (2014). Automatic recognition of serial numbers in bank notes. *Pattern Recognition*, *47*(8), 2621-2634.

Fodor, I. K. (2002). A survey of dimension reduction techniques. LLNL Technical Report, UCRL-ID-148494

Fouladi, K., Araabi, B., & Kabir, E. (2013). A fast and accurate contour-based method for writer-dependent offline handwritten Farsi/Arabic sub-words recognition. *International Journal on Document Analysis and Recognition.* 1-23.

Fradkin, D., & Madigan, D. (2003). Experiments with random projections for machine learning. *The ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 517-522.

Gesualdi, D.R., & Seixas, J.M. (2002). Character recognition in car license plates based on principal components and neural processing. *The VII Brazilian Symposium on Neural Networks*, pp. 206-211.

Gharoie Ahangar, R., & Farajpoor Ahangar , M. (2009). Handwritten Farsi Characters Recognition using Artificial Neural Network. *International Journal of Computer Science and Information Security, 4* (2).

Ghods, V., & Kabir, E. (2013a). A Study on Farsi Handwriting Styles for Online Recognition. *International Malaysian Journal of Computer Science*, *26*(1), 44-59.

Ghods, V., & Kabir, E. (2013b). Effect of delayed strokes on the recognition of online Farsi handwriting. *Pattern Recognition Letter*, *34(*5), 486-491.

Ghods, V., & Kabir, E. (2013c). Decision fusion of horizontal and vertical trajectories of online Farsi sub-words. *Engineering Applications of Artificial Intelligence, 26(*1), 544-550.

Gonzalez, R.C., Woods, R.E., & Eddins, S.L. (2009). *Digital Image Processing using MATLAB*. 2nd ed. *Gatesmartk Publishing*.

Gouda, A.M., & Rashwan, M.A. (2004). Segmentation of connected Arabic characters using hidden Markov models. *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications,* pp. 115-119.

Gupta, M.R., Jacobson, N.P., & Garcia, E.K. (2007). OCR binarization and image pre-processing for searching historical documents. *Pattern Recognition (40)*, 389-397.

Guyon, I., & Elisseeff , A. (2003). An Introduction to variable and feature selection. *Journal of Machine Learning Research, 3*(1), 1157-1182.

Haghighi, P.J., Nobile, N., He, C.L., & Suen, C.Y. (2009). A new large-scale multi-purpose handwritten Farsi database. *Lecturer Notes in Computer Science*, *5627*, pp. 278–286. Springer Berlin Heidelberg.

Halavati, R., & Shouraki, S.B. (2007). Recognition of Persian online handwriting using elastic fuzzy pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence, 21* (3), 491-513.

Hamid, A., & Haraty, R. (2001). A neuro-heuristic approach for segmenting handwritten Arabic text. *IEEE International Conference on Computer System and Applications*, pp. 110-113.

Hanmandlu, M., Murali Mohan, K.R., Chakraborty, S., Goyal, S., & Roy Choudhury, D. (2003). Unconstrained handwritten character recognition based on fuzzy logic. *Pattern Recognition*, *36,* 603-623.

Harifi, A., & Aghagolzadeh, A. (2005). A New Pattern for Handwritten Persian/Arabic Digit Recognition. *World Academy of Science, Engineering and Technology, 3,* 249–252.

Harouni, M ., Mohamad, D., & Rasouli, A. (2010). Deductive Method for Recognition of Online Handwritten Persian/Arabic Characters. *2nd IEEE International Conference on Computer and Automation Engineering, 5,* pp. 791-795.

http://www.cedar.buffalo.edu/Datasets/jocr

Huang, L., Wan, G., & Liu, C. (2003). An Improved Parallel Thinning Algorithm. *7th International Conference on Document Analysis and Recognition, 3,* pp. 780-783.

Hyun-chul, K. Daijin, K., & Sung Yang, B. (2002). A numeral character recognition using the PCA mixture model. *Pattern Recognition Letters, 23*, 103-111.

Impedovo, S. (2014). More than twenty years of advancements on Frontiers in Handwriting Recognition. *Pattern Recognition*, *47* (3), 916-928.

Impedovo, S., Mangini, F. M., & Barbuzzi, D. (2014). A novel prototype generation technique for handwriting digit recognition. *Pattern Recognition*, *47*(3), 1002-1010.

Izabatene, H.F., Benhabib, W., & Ghardaoui, S. (2010). Contribution of kernels on the SVM performance. *International Journal of Applied Science. 10*, 831–836.

Izakian, H., Monadjemi, S.A., Tork Ladani, B., & Zamanifar, K. (2008). Multi Font Farsi/Arabic Isolated Characters Recognition Using Chain Codes. *World Academy of Science, Engineering and Technology*, *43*, 67-70.

Javed, I., Ayyaz, M.N., & Mehmoud, W. (2007). Efficient Training Data Reduction for SVM based Handwritten Digits Recognition. *IEEE International Conference on Electrical Engineering*, pp. 1-4.

Jenabzade, M.R., Azmi, R., Pishgoo, B., & Shirazi, S. (2011). Two Methods for Recognition of Handwritten Farsi Characters. *International Journal of Image Processing*, *5*(4), 512-519.

Jumari, K., & A.Ali, M. (2002). A Survey and Comparative Evaluation of Selected Off-line Arabic Handwritten Character Recognition systems. *Journal Teknologis 36(E), Universiti Teknologi Malaysia*, 1–18.

Kabir, E. (2009). Evolution of OCR and the state-of-the-art of Farsi OCR. *in OCR Research Group, Research Guide to Farsi OR, Iran Supreme Council of Information and Communication Technology*, pp. 39-76 (in Farsi)

Kamranian, Z., Monadjemi, S. A., & Nematbakhsh, N. (2013). A novel free format Persian/Arabic handwritten zip code recognition system. *Computers & Electrical Engineering*, *39*(7), 1970-1979.

Kanungo, T., Marton, G . A . , & Bulbul, O . (1999). OmniPage vs. Sakhr: paired model evaluation of two Arabic OCR products. *Proceeding of International Conference in Document Recognition and Retrieval*, *6*, pp. 109-121.

Kapour, R., Bagai, D., & Kamal, T.S. (2004). A new algorithm for skew detection and correction**,** *Pattern Recognition Letters, 25*, 1215–1229.

Karic, M., & Martinovic, G. (2013). Improving Offline Handwritten Digit Recognition Using Concavity-Based Features. *International Journal of Computers, Communications & Control*, *8*(2), 220-234.

Keogh, E.J., & Pazzani, M.J. (2000). A simple dimensionality reduction technique for fast similarity search in large time series databases. *Knowledge Discovery and Data Mining, Current Issues and New Applications,* Springer Berlin Heidelberg, pp. 122-133.

Khalifa, M., Bingru, Y., & Mohammed, A. (2011). A Robust SIFT Feature for Fast Offline Arabic Words Classification. *IEEE International Conference on Computer Science and Automation Engineering, 4,* pp. 83-86.

Khan, K.U., & Haider, I. (2010). Online recognition of multi stroke handwritten Urdu characters. *International Conference on Image Analysis and Signal Processing*. pp. 284-290.

Khedher, M., & Abandah, G. (2002). Arabic Character Recognition using Approximate Storke Sequence. *3rd international Conference on Language Resources and Evaluation,*

Khedher, M.Z., Abandah, Gh.A., & Al-Khawaldeh, A.M. (2005). Optimizing Feature Selection for Recognizing Handwritten Arabic Characters. *World Academy of Science, Engineering and Technology (4),* 81-84.

Kherallah, M., Elbaati, A., Abed, H., & Alimi, A.M. (2008). The on/off (LMCA) dual Arabic handwriting database. *11th International Conference on Frontiers in Handwriting Recognition*.

Kheyrkhah, A.R., & Rahmanian, E. (2007). Optimizing a Farsi handwritten character recognition system by selecting effective features on classifier using genetic algorithm. *First Joint Congress on Fuzzy and Intelligent Systems, 43-52 (in Persian).*

Khorsheed, M.S. (2002). Off-line Arabic character recognition – a review. *Pattern Analysis and Applications, 5*(1), 31-45.

Khorsheed, M.S. (2007) Offline recognition of omni font Arabic text using the HMM ToolKit (HTK). *Pattern Recognition Letters, 28*(12), 1563-1571.

Khosravi, H., & Kabir, E. (2007). Introducing a very large dataset of handwritten Farsi digits and a study on their varieties. *Pattern Recognition Letters, 28*(10), 1133-1141.

Khosravi, S., Razzazi, F., Rezaei, H., & Sadigh, M.R. (2007). A Comprehensive Handwritten Image Corpus of Isolated Persian/Arabic Characters for OCR Development and Evaluation. *9th International symposium on signal processing and its application,* pp. 1-4.

Khosravi, H., & Kabir, E. (2009). A blackboard approach towards integrated Farsi OCR system. *International Journal of Design and Innovation Research,* 21-32.

Kim, H.C., Kim, D., & Yang Bang, S. (2002). A numeral character recognition using the PCA mixture model. *Pattern Recognition Letters*, 23, 103-111.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence, 14*(2), pp. 1137-1145.

Kuri-Morales, A., & Rogriguez-Erazo, F. (2009). A search space reduction methodology for data mining in large databases. *Engineering Applications of Artificial Intelligence, 22*(1), pp. 57-65.

Le Cun, Y., Bottou, L., Bengio, Y., & Haffiner, P. (1998). Gradient based learning applied to document recognition, *Proceeding IEEE* , 86(11), pp. 2278-2324.

Le, H. M., Duong, A. T., & Tran, S. T. (2013). Multiple-Classifier Fusion Using Spatial Features for Partially Occluded Handwritten Digit Recognition. In *Image Analysis and Recognition,* pp. 124-132. Springer Berlin Heidelberg.

Li, Sh., Fevenes, T., Krzyzak, A., & Li, S. (2006). Automatic clinical image segmentation using pathological modeling, PCA and SVM. *Engineering Applications of Artificial Intelligence, 19*(4), 403-410.

Lorigo, L.M., & Govindaraju, V. (2005). Segmentation and Pre-Recognition of Arabic Handwriting. *IEEE International Conference on Document Analysis and Recognition*, pp. 605–609.

Lorigo, L.M., & Govindaraju, V. (2006). Offline Arabic handwriting recognition: a survey. *IEEE Transaction Pattern Analysis and Machine Intelligence*, *28*(5), 712-724.

Lu, Y., & Tan, C.L. (2003). A Nearest neighbor chain based approach to skew estimation in document images. *Pattern Recognition Letters, 24*, 2315-2323.

Mahmoud, S.A., & Mahmoud, A.S. (2006). Arabic Character Recognition using Modified Fourier Spectrum (MFS). *IEEE Proceeding of the Geometric Modeling and Imaging – New Trends,* pp. 155-159.

Mahmoud, S. A., & Olatunji, S. O. (2010). Handwritten Arabic numerals recognition using multi-span features & Support Vector Machines. 1*0th IEEE International Conference on Information Sciences Signal Processing and their Applications,* pp. 618-621.

Mandal, R.K., & Manna, N.R. (2011). Handwritten English Character Recognition using Row-wise Segmentation Technique (RST). *International Journal of Computer Applications*, 5-9.

Manjunath Aradhya, V.N., Hemantha Kumar, G., & Noushath, S. (2008). Multilingual OCR system for South Indian scripts and English documents: An approach based on Fourier transform and principal component analysis. *Engineering Applications of Artificial Intelligence*, 21(4), 658-668.

Marti, U.V., Bunke, H. (2002). The IAM-database: An English Sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition, 5*(1), 39-46.

Mehran, R., Shali, A., & Razzazi, F. (2004). A statistical correction-rejection strategy for OCR outputs in Persian personal information forms. *2nd International Conference on Information Technology for Applications*, pp. 247-249.

Mehran, R., Pirsiavash, H., & Razzazi, F. (2005). A front-end OCR for Omni-font Persian/Arabic cursive printed documents. *Proceedings of the Digital Imaging Computing: Techniques and Applications,* pp. 56-60.

Menhaj, M.B., & Adab, M. (2002). Simultaneous segmentation and recognition of Farsi/Latin printed texts with MLP. *IEEE International Joint Conference on Neural Networks*, *2*, pp. 1534-1539.

Milone, D.H., Stegmayer, G., Kamenetzky, L., LóPez, M., & Carrari, F. (2013). Clustering biological data with SOMs: On topology preservation in non-linear dimensional reduction. *Expert System with Applications, 40*, 3841-3845.

Mirsharif, G. Badami, M., Salehi, B. and & Azimifar, Z. (2012). Recognition of Farsi Handwritten Digits using a Small Feature Set. *International Journal of Computer and Electrical Engineering, 4*(4), 588-591.

The MNIST dataset is available at http://yann.lecun.com/exdb/mnist/index.html.

Moradi, M., Poormina, M.A., & Razzazi, F. (2009a). FPGA Implementation of Feature Extraction and MLP Neural Network Classifier for Farsi Handwritten Digit Recognition. *3rd IEEE European Symposium on Computer Modeling and Simulation*, pp. 231-234.

Moradi, M., Poormina, M.A., & Razzazi, F. (2009b). Implementation of An Accurate Farsi Handwritten Digit Recognition System On FPGA. *2rd IEEE Asia-Pacific Conference on Computational Intelligence and Industrial Applications*, pp. 401-404.

Moradi, M., Poormina, M.A., & Razzazi, F. (2010). A New Method of FPGA Implementation of Farsi Handwritten Digit Recognition. *European Journal of Scientific Research, 39* (3), 309-315.

Mousavinasab, Z., & Bahadori, H. (2012). The presentation a new method based slope variations and DTW algorithm for recognition of Farsi handwritten digits, *International Journal of science and Advanced Technology, 2*(2), 92-94.

Mowlaei, A., Faez, K., & Haghighat, A.T. (2002). Feature Extraction with Wavelet Transform for Recognition of Isolated Handwritten Farsi/Arabic Characters and Numerals. *14th IEEE International Conference on Digital Signal Processing, 2,* pp. 923-926.

Mowlaei, A., & Faez, K. (2003). Recognition of isolated handwritten Persian/Arabic Characters and Numerals using Support Vector Machines. *IEEE International Workshop on Neural Networks for Signal Processing,* pp. 547-554.

Mozaffari, S., Faez, K., & Rashidy Kanan, H. (2004a). Recognition of isolated handwritten Farsi/Arabic alphanumeric using fractal codes. *Proceedings of 6th IEEE Southwest Symposium on Image Analysis and Interpretation, 6,* pp. 104-108.

Mozaffari, S., Faez, K., & Rashidy Kanan, H. (2004b). Feature Comparison between Fractal Codes and Wavelet Transform in Handwritten Alphanumeric Recognition using SVM Classifier. *IEEE International Conference on Pattern Recognition, 2*, pp. 331-334.

Mozaffari, S., Faez, K., & Ziaratban, M. (2005a). A Hybrid Structural/Statistical classifier for Handwritten Farsi/Arabic Numeral Recognition. *International Conference on Machine Vision Application,* pp. 211–218.

Mozaffari, S., Faez, K., & Ziaratban, M. (2005b). Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters. *8th IEEE International Conference on Document Analysis and Recognition*, pp. 237–241.

Mozaffari, S., Faez, K., & Ziaratban, M. (2005c). Character Representation and Recognition using Quad tree-based Fractal Encoding Scheme. *8th IEEE International Conference on Document Analysis and Recognition*, pp. 819-823.

Mozaffari, S., Faez, K., & Rashidy Kanan, H. (2005). Performance Evaluation of Fractal Feature in Recognition of Postal codes using an RBF Neural Network and SVM Classifier. *International Conference on Machine Vision Application*, pp. 562–565.

Mozaffari, S., Faez, K., Faradji, F., Ziaratban, M., & Golzan, S.M. (2006). A Comprehensive Isolated Farsi/Arabic Character Database for Handwritten OCR Research. *10th International Workshop on Frontiers in Handwriting Recognition*, pp. 385-389. http://ele.aut.ac.ir/imageproc/downloads/IFHCDB.html

Mozaffari, S., Faez, K., Margner, V., & El-Abed, H. (2007). Strategies for Large Handwritten Farsi/Arabic Lexicon Reduction. *9th IEEE International Conference on Document Analysis and Recognition, 1*, pp. 98-102.

Mozaffari, S., El-Abed, H., Margner, V., Faez, K., & Amirshahi, A. (2008). IfN/Farsi – Database: A database for Farsi Handwritten City Names. *International Conference on Frontiers in Handwriting Recognition*.

Mozaffari, S., Faez, K., Margner, V., & El-Abed, H. (2008). Lexicon reduction using dots for offline Farsi/Arabic handwritten word recognition. *Pattern Recognition Letters, Vol. 29, no. 6*, pp. 1-11.

Nabavi, S.H., Ebrahimpour, R., & Kabir, E. (2005). Recognition of handwritten Farsi digits using classifier combination. *The 3rd Conference on Machine Vision, Image Processing and Application*, pp. 116-119 (in Farsi).

Nagabhushan, P., & Alaei, A. (2009). Unconstrained Handwritten Text-line Segmentation Using Morphological Operation and Thinning Algorithm, *IICAI*. Pp. 2080-2091.

Nagabhushan, P., & Alaei, A. (2010). Tracing and Straightening the Baseline in Handwritten Persian/Arabic Text Line : A New Approach Based on Painting technique. *International Journal of Computer Science and Engineering, 2*(4), 907-916.

Nasrollahi, S., & Ebrahimi, A. (2011). Wavelet transform for skew angle detection in printed Persian documents. *3rd International Conference on Digital Image Processing, Appril*, Doi:10.1117/12.896301.

Nasrudin, M. F., Omar, K., Zakaria, M. S., & Yeun, L. C. (2008,). Handwritten cursive Jawi character recognition: A survey. 5th *IEEE International Conference on Computer Graphics, Imaging and Visualization*, pp. 247-256.

Nawaz, S.N., Sarfaraz, M., Zidouri, A., & Al-khatib, W.G. (2003). An approach to offline Arabic character recognition using neural networks. *10th IEEE International* *conference on Electronics, Circuits and Systems, 3,* pp. 1328-1331.

Noaparast, K., & Broumandnia, A. (2009). Persian Handwritten Word Recognition Using Zernike and Fourier-Mellin Moments. *IEEE 5th International Conference on Sciences of Electronics, Technologies of Information and Telecommunication,* pp. 1-7.

Nourouzian, E., Mezghani, N., Mitiche, A., & Robert, B. (2006). Online Persian/Arabic character recognition by polynomial representation and a Kohonen network. *IEEE International Conference on Pattern Recognition,* pp. 222-226.

Olivier. G., Miled, H., Romeo, K., & Lecourtier, Y. (1997). Segmentation and Coding of Arabic Handwritten Words. *13th IEEE International Conference on Pattern Recognition,* pp. 264-268.

Omidyeganeh, M., Nayebi, K., Azmi, R., & Javadtalab, A. (2005). A New Segmentation technique for Multi Font Farsi/Arabic Texts. *IEEE International Conference on Acoustics Speech and Signal Processing, 2,* pp. 757-760.

Omidyeganeh, M., Azmi, R., Nayebi, K., & Javadtalab, A. (2007). A new method to improve multi font Farsi/Arabic character segmentation results: using extra classes of some character combinations. *International Conference of Advances in Multimedia Modeling, 1*, pp. 670-679.

Pal. U., Jayadevan, R., & Sharma, N. (2012). Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques. *ACM Transactions on Asian Language Information Processing, 11*(1), 1-12.

Pan, W.M., Bui, T.D., & Suen, C.Y. (2009). Isolated Handwritten Farsi Numerals Recognition Using Sparse and Over-Complete Representations. *10th IEEE International Conference on Document Analysis and Recognition*, pp. 586–590.

Parhami, B., & Taraghi, M. (1981). Automatic recognition of printed Farsi Texts. *Pattern Recognition, 14*(6), 395-401.

Parvez, M.T., & Mahmoudi, S.A. (2013). Offline Arabic Handwritten Text Recognition : A Survey. *ACM Computing Survey, 45*(2), 23.

Patel, C.I., Patel, R., & Patel, P. (2011). Handwritten Character Recognition using Neural Network. *International Journal of Scientific & Research, 2*(5), 1-6.

Pechwitz, M., & Margner V. (2003). HMM based approach for handwritten Arabic word recognition using the IFN/ENIT- database. *International Conference on Data Analysis and Recognition*, pp. 890-894.

Peng, X., Cao, H., Setlur, S., Govindaraju, V., & Natarajan, P. (2013). Multilingual OCR research and applications: an overview. *4th International Workshop on Multilingual OCR* (p. 1). ACM, 2013.

Pengl, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transaction on PAMI, 27*(8), 1226-1238.

Phinyomark, A., Phukpattaranont, P., & Limsakul, Ch. (2012). Feature reduction and selection for EMG signal classification. *Expert Systems with Applications,* 39, 7420-7431.

Pirsiavash, H., Mehran, R., & Razzazi, F. (2005). A Robust Free Size OCR for Omni-font Persian/Arabic Printed Document using Combined MLP/SVM. *Lecturer Notes in Computer Science, 3773,* pp. 601-610. Springer Berlin Heidelberg.

Pourasad, Y., Hassibi, H., & Banaeyan, M. (2011). Persian Characters Recognition Based on Spatial Matching. *World Applied Sciences Journal, 13*(2), 239-243.

Pradeep, J., Srinivasan, E., & Himavathi, S. (2011). Neural Network based Handwritten Character Recognition System without Feature Extraction. *IEEE International Conference on Computer, Communication and Electrical Technology*, pp. 40-44.

Rafaa, A.D., & Nordin J. (2011). Offline OCR System for Machine Printed Turkish using Template Matching. *Journal of Advanced Materials Research, (341-342),* 565-569.

Rajabi, M., Nematbakhsh, N., & Monadjemi, A.H. (2012). A New Decision Tree for Recognition of Persian Handwritten Characters. *International Journal of Computer Applications, (44)*6, 52-58.

Rasheed, N.A. (2011). Neural Network Based Segmentation Algorithm for Arabic Character Recognition. *Journal of Babylon University / Pure and Applied Sciences, 19*(3), 823-828.

Rashnodi, O., Sajedi, H., & Saniee Abadeh, M. (2011). Persian Handwritten Digit Recognition using Support Vector Machines. *International Journal of Computer Applications, (29)*12, 1-6.

Sabri, A.M., & Sunday, O.O. (2010). Handwritten Arabic numerals recognition using multi span features & support vector machines. *International Conference on Information Science, Signal Processing and their Applications,* pp. 618–621.

Sadri, J., Suen, C.Y., & Bui, T.D. (2003). Application of Support Vector Machines of Handwritten Arabic/Persian Digits. *Proceedings of 2th Iranian Conference on Machine Vision and Image Processing*, *1*, pp. 300-307.

Sadri, J., Izadi, S., Solimanpour, F., Suen, C.Y., & Bui, T.D. (2007). State-of-the-art in Farsi script recognition. *IEEE 9th International Symposium on Signal Processing and its Applications*, pp. 1-6.

Safabakhsh, R., & Khadivi, SH. (2000). Document Skew Detection Using Minimum-Area Bounding Rectangle. *IEEE International Conference on Information Technology: Coding and Computing*, pp. 253-258.

Safabakhsh, R., & Adibi, P. (2005). Nastaaligh handwritten word recognition using a continuous-density variable-duration HMM. *Arabian Journal for Science and Engineering*, *30*(1B), 95-118 .

Salehpour, M., & Behrad, A. (2010). Cluster Based Weighted SVM for the Recognition of Farsi Handwritten Digits. *10th IEEE Symposium on Neural Network Applications in Electrical Engineering*, pp. 219-223.

Salmani Jelodar, M ., Fadaeieslam, M.J., & Mozayani, N. (2005). A Persian OCR System using Morphological Operators. *World Academy of Science, Engineering and Technology, 4*, 137-140.

Samimi Daryoush, K., Khademi, M., Nikookar, A., & Farahani, A. (2010). The Application of Local Linear Neuro Fuzzy Model in Recognition of Online Persian Isolated Characters. *3rd International Conference on Advanced Computer Theory and Engineering*. pp. 574-577.

Sanaei, Z., Abolfazli, S., Gani, A., & Buyya, R. (2013). Heterogeneity in mobile cloud computing: taxonomy and open challenges. *IEEE Communications Surveys & Tutorials,* 99, 1-24.

Saracoglu, R. (2012). Hidden Markov model-based classification of heart valve disease with PCA for dimension reduction. *Engineering Applications of Artificial Intelligence*, *25*(7), 1523-1528.

Sarfaraz, M., Nawaz, S.N., & Al-Khuraidly, A. (2003). Offline Arabic text recognition system. *IEEE International Conference on Geometric Modeling and Graphics*, pp. 30-36.

Sari, T., & Sellami, M. (2007). Overview of Some Algorithms of Off-Line Arabic Handwriting Segmentation. *International Arab Journal of Information Technology, 4(4)*, 289-300.

Sari, T., Souici, L., & Sellami, M. (2002). Off-Line Handwritten Arabic Character Segmentation Algorithm: ACSA. *8th IEEE International Workshop on Frontiers in Handwriting Recognition*, pp. 452–457.

Shah, M., & Jethava, G.B. (2013). A Literature Review On Hand Written Character recognition. *Indian Streams Research Journal, 3*(2), 1-19, 2013.

Shahabi, A. S., & Kangavari, M. R. (2007). A Fuzzy Approach for Persian Text Segmentation Based on Semantic Similarity of Sentences. In *Intelligent Information Processing III,* pp. 411-420, Springer US.

Shanbehzadeh, J., Pezashki, H., & Sarrafzadeh, A. (2007). Feature Extraction from Farsi Handwritten Letters. *Image and Vision Computing,* 35-40.

Sharma, A., & Palimal, K.K. (2008). Fast principal component analysis using fixed-point algorithms. *Pattern Recognition Letters*, 28, 1151-1155.

Sharma, O.P., Ghose, M.K., Bikram Shah, K., & Kumar Thakur, B. (2013). Recent trends and tools for feature extraction in OCR technology. *International Journal of Soft Computing and Engineering*, *2*(6), 220-223.

Shayegan, M.A., & Aghabozorgi, S. (2014a). A New Dataset Size Reduction Approach for PCA-Based Classification in OCR Application, *Mathematical Problems in Engineering, vol. 2014,* Article ID 537428, 14 pages, 2014. doi:10.1155/2014/537428.

Shayegan, M.A., & Aghabozorgi, S. (2014b). A New Method for Arabic/Farsi Numeral Dataset Size Reduction via Modified Frequency Diagram Matching, *Kybernetes, 43*(5), 817-834.

Shayegan, M.A., Aghabozorgi, S., & Raj, R.G. (2014). A Novel Two-Stage Spectrum-Based Approach for Dimensionality Reduction: A Case Study on the Recognition of Handwritten Numerals, *Journal of Applied Mathematics, vol. 2014,* Article ID 654787, doi:10,1155/2014/654787.

Shayegan, M.A., & Cha, Ch.S. (2012). A New Approach to Feature Selection in
Handwritten Farsi/Arabic Character Recognition, *IEEE International Conference on Advanced Computer Science Applications and Technologies (ACSAT),* pp. 506-511.

Sheikh, N.A., Ali Mallah, Gh., & Shaikh, Z.A. (2009). Character Segmentation of Sindhi, an Arabic Style Scripting Language, using Height Profile Vector, *Australian Journal of Basic and Applied Science, 3(*4*),* 4160-4169.

Shirali, S., Manzuri, M.T., & Shirali, M. H. (2006a). Preparing Persian/Arabic Scanned Images for OCR. *2nd IEEE International Conference on Information and Communication Technologies, 1,* pp. 1332-1336.

Shirali, S., Manzuri, M.T., & Shirali, M.H. (2006b). Page segmentation of Persian/Arabic printed text using ink spread effect. In IEEE *International Joint Conference on SICE-ICASE,* pp. 259-262.

Shirali, S., Manzuri, M.T., & Shirali, M. H. (2007). A Skew Resistant Method for Persian Text Segmentation. *IEEE Symposium on Computational Intelligence in Image and Signal Processing,* pp. 115-120.

Shirali, M.H., & Shirali, S. (2008). Removing noises similar to dots from Persian scanned documents. IEEE *International Colloquium on Computing, Communication, Control, and Management, 2,* pp. 313-317.

Singh, D., Singh, S., & Dutta, M. (2010). Handwritten Character Recognition using Twelve Directional Feature Input and Neural Network. *International Journal of Computer Applications, 1*(3), 82-85.

Sitamahalakshmi, T., Vinay Babu, A., & Jagadeesh, M. (2010). Character Recognition using Dempster-Shafer Theory – Combining Different Distance Measurement Methods. *International Journal of Engineering and Technology, 2* (5), 1177-1184.

Slimane, F., Ingold, R., Kanoun, S., Alimi, A. M., & Hennebert, J. (2009). A new Arabic printed text image database and evaluation protocols. *10th IEEE International Conference on Document Analysis and Recognition*, July, pp. 946-950.

Soleymani, M., & Razzazii, F. (2003). An Efficient Front-End System for Isolated Persian/Arabic Character Recognition of Handwritten Data – Entry Forms. *Journal of Computational Intelligence, 1,* 193-196.

Solimanpour, F., Sadri, J., & Suen, C. (2006). Standard databases for recognition of handwritten digits, numerical string, legal amounts, letters and dates in Farsi language. *10th International Workshop on Frontiers in Handwriting Recognition*, pp. 3-7.

Soltanzadeh, H., & Rahmati, M. (2004). Recognition of Persian handwritten digits using image profiles of multiple orientations. *Pattern Recognition Letters, 25*(14), 1569-1576.

Song, M., Yang, H., Siadat, S.H., Pechenizkiy, M. (2013). A comparative study of dimensionality reduction techniques to enhance trace clustering performances. *Expert Systems with Applications*, 40, 3722-3737.

Timsari, B., & Fahimi, H. (1996). Morphological approach to character recognition in machine-printed Persian words. In P*roceeding of the International Society for Optical Engineering, 2660* , pp. 184-191 .

Toosizadeh, N., & Eshghi, M. (2005). Design and Implementation of a New Persian Digits OCR Algorithm in FPGA Chips. *13th European signal conference,*

Tsai, F.S. (2011). Dimensionality reduction techniques for blog visualization. *Expert System with Applications, 38*, 2766-2773.

Urmanov, A.M., Bougaev, A.A., & Gross, K.C. (2007). Reducing the Size of a Training Set for Classification. *US Patent Application Publication*, 2381–2385.
http://www.freepatentsonline.com/y2007/0260566.html

Van der Maaten, L.J.P., Postma, E.O. & Van Den Herik, H.J. (2009). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, *10*(1-41), 66-71.

Vaseghi, B., Alirezaee, S., Ahmadi, M., & Amirfattahi, R. (2008). Off-line Farsi/Arabic Handwritten Word Recognition Using Vector Quantization and Hidden Markov Model. *IEEE International Conference on Multi Topic*, pp. 575-578.

Vishwanathan, S.V.N., & Murty, M.N. (2004). Use of Multi category Proximal SVM for Data Set Reduction. *International Journal of Studies in fuzziness and soft computing, 140*, 3–20.

Xiu, P., Peng, L., Ding, X., & Wang, H. (2006). Offline Handwritten Arabic Character Segmentation with Probabilistic Model. *Springer Berlin/Heidelberg, 3872,* 402-412.

Yang, C., Shuhua, W., & Heng, L. (2002). Skew detection and correction in document images based on straight-line fitting. *Pattern Recognition Letters*, *24*, 1871–1879.

Yang, Ch., Zhang, W., Zou, J., Hu, Sh., & Qiu, J. (2013). Feature selection in decision systems: A Mean Variance approach. *Journal of Mathematical Problems in Engineering*, 2013.

Yousefi, H., Faez, K., & Ardekani, M.H. 2012. A novel approach of skew estimation and correction in Persian manuscript text using Radon transform. *IEEE Symposium on Computers & Informatics,* pp. 198-202.

Zand, M., Naghsh Nilchi, A., & Monadjemi, A. (2008). Recognition based Segmentation in Persian Character Recognition. *International Journal of Computer and Information Science and Engineering, 2*(1), 14-18.

Zeki, A.M. (2005). The Segmentation Problem in Arabic Character Recognition, The State of the Art. The f*irst International conference on Information and Communication Technologies*. pp. 11-26.

Zhang, P., Suen, C.Y., & Bui, T.D. (2004). Multi-modal nonlinear feature extraction for the recognition of handwritten numerals. *Proceedings of the first Canadian conference on computer and robot vision*, pp. 393–400.

Zheng, L., Hassin, A.H., & Tang, X. (2004). A new algorithm for machine printed Arabic character segmentation. *Pattern Recognition Letters*, *25,* pp. 1723-1729.

Zhongdong, W., Jianping, Y., Weixin, X., & Xinbo, G. (2004). Reduction of Training Datasets via Fuzzy Entropy for Support Vector Machines. *IEEE International Conference on Systems, Man and Cybernetics, 3,* pp. 2381–2385.

Ziaratban, M., Faez, K., & Faradji, F. (2007). Language-Based Feature Extraction using Template Matching in Farsi/Arabic Handwritten Numeral Recognition. *9th IEEE International Conference on Document Analysis and Recognition,* pp. 297-301.

Ziaratban, M., Faez, K., & Ezoji, M. (2007). Use of Legal Amount to Confirm or Correct the Courtesy Amount on Farsi Bank Checks. *IEEE International Conference on Document Analysis and Recognition, 2,* pp. 1123-1127.

Ziaratban, M., & Faez, K. (2008). A Novel Two-Stage Algorithm for Baseline Estimation and Correction in Farsi and Arabic handwritten Text Line. *IEEE International Conference on Pattern Recognition,* pp. 1-5.

Ziaratban, M., Faez, K., & Allahveiradi, F. (2008). Novel Statistical Description for the Structure of Isolated Farsi/Arabic Handwritten Characters. In *Proceeding of 11th ICFHR,* pp. 332-337.

Ziaratban, M., Faez, K., & Bagheri, F. (2009). FHT: An Unconstraint Farsi Handwritten Text Database. *10th IEEE International Conference on Document Analysis and Recognition,* pp. 281–285.

Ziaratban, M., & Faez, K. (2009). Non-Uniform slant estimation and correction for Farsi/Arabic Handwritten Words. *International Journal on Document Analysis and Recognition, 12*, 249–267.

Ziaratban, M., & Faez, K. (2012). Detection and compensation of undesirable discontinuities within the Farsi/Arabic subwords. *International Arab Journal of Information Technology*, *8*(3), 293-301.

Publications

1) Shayegan, M.A., & Aghabozorgi, S. (2014). A New Dataset Size Reduction Approach for PCA-Based Classification in OCR Application, *Mathematical Problems in Engineering (ISI Cited, Q2), vol. 2014,* Article ID 537428, 14 pages, 2014. doi:10.1155/2014/537428.

2) Shayegan, M.A., Aghabozorgi, S., & Raj, R.G. (2014). A Novel Two-Stage Spectrum-Based Approach for Dimensionality Reduction: A Case Study on the Recognition of Handwritten Numerals*, Journal of Applied Mathematics (ISI Cited, Q2), vol. 2014,* Article ID 654787, doi:10,1155/2014/654787.

3) Shayegan, M.A., & Aghabozorgi, S. (2014). A New Method for Arabic/Farsi Numeral Dataset Size Reduction via Modified Frequency Diagram Matching, *Kybernetes (ISI Cited, Q4), 43*(5), 817-834.

4) Shayegan, M.A., & Cha, Ch.S. (2012). A New Approach to Feature Selection in Handwritten Farsi/Arabic Character Recognition, *IEEE International Conference on Advanced Computer Science Applications and Technologies (ACSAT),* pp. 506-511.

5) Shayegan, M.A., Aghabozorgi, S., & Raj, R.G. (2014). Ensemble of Decision Stumps for Handwritten Farsi/Arabic Digit Recognition, *Abstract and Applied Analysis (ISI Cited, Q2)*, Under review.

Appendix I

OCR Software Supporting Farsi Language

Although the progress in developing FOCR systems has been slower than OCR systems for Latin, there have been some successes in developing techniques for recognizing printed Farsi text. For example, FOCR software, ARAX2, has been designed basically as a commercial product and has achieved acceptable level of performance of 99.5% accuracy in recognizing multi-font – multi-size printed Farsi texts. Also, the Omnipage software has achieved 86% accuracy in recognizing clean Farsi text images. Until now, however, no fully operational commercial product with acceptable performance in recognizing handwritten Farsi manuscripts is available in the market. The main reason for this is related to the diversity of writing styles in handwritten documents.

Table 1 shows some of the available OCR systems in the market. Most of these products have been designed to recognize non-Farsi scripts. However, they support printed Farsi documents with moderate level of accuracy. The reported performances in the last two columns have been extracted from the relevant companies' websites.

Table 1: Some available OCR software with supporting Farsi alphabet

Product Name	Company Provider	Original Language	Performance for Original Language	Performance for Farsi
FineReader ver. 9.0	ABBYY Software Company	Multi. L.	-------------	Less than 60%
Al_Qari ver. 2.0	ALAlamiah (language Source LTD.)	Arabic	99.0%	-------------
ARAX2	HODA_soft company	Farsi	99.5%	99.5%
Sakhr utomatic Reader	Sakhr company	Arabic	90.33%	60% to 70%
CiyaOCR	Ciasoft company	Arabic	-------------	-------------
ICRA 4.0 Arabic	Arab Scientific Software & Engineering Technologies	Arabic	Less than 90%	-------------
NovoVerus	NovoDynamics company	Multi L.	94.5%	------------
OmniPage Prof. ver. 18.1	Recognita (OCR System GmbH)	Multi. L.	99%	85%
ReadIris Pro (ver12.57 ME)	Iris company	Arabic	84.0%	60% to 70%

Majority of mentioned software in Table 1.1 have been designed to recognize scripts of non-Farsi documents. However, they support printed Farsi and Arabic documents with moderate accuracy, too. But, they do not achieve to an acceptable recognition rate for handwritten Farsi texts.

Although the progress in developing FOCR is slower than Latin and Chinese, but there has been some valuable success in developing solution for recognizing printed Farsi text (Khosravi and Kabir, 2009). Nowadays, there is a little, but very good commercial products for printed Farsi OCR with more than 99.5% accuracy. Among above products, ARAX2 has basically designed as a commercial product with acceptable performance for recognizing printed Farsi text. Also, ReadIris software achieves to 84% accuracy for clean images for Arabic text. However, until now, there is not a commercial product with acceptable performance for recognizing handwritten Farsi documents. The main reason for this deficiency is related to nature of handwritten Farsi manuscripts. Generally, in all languages, diversity of handwritten documents is very more than printed documents, because every person has his/her own writing style. This fact causes handwritten OCR operations be one of the hardest operation in PR area.

Although Farsi alphabet is very similar to Arabic alphabet, but there are minor, but important, differences between these two alphabets and their styles. Duo to these differences, a handwritten Arabic OCR system might not perform well for Farsi and vice versa. The reported results in Table 1.1 show this fact. Some of these differences are:

- There is one letter "ى" in Arabic language which is not included in Farsi alphabet.
- Farsi alphabet has four extra letters " گ ", " چ ", " پ " and " ژ " in comparison to Arabic alphabet. These four letters have 14 different forms based on their locations

in a word. Therefore they add 14 extra classes to pattern space in Farsi OCR systems.

- There are a large number of words in Farsi language which they are not in Arabic language and vice versa. Therefore an OCR system will face to serious problems in holistic approach which recognition is carried without segmentation, because in this approach, systems use a dictionary (or a lexicon) for checking the final result.
- There are some writing styles only in Farsi (such as Shekaste-nasta'aligh style) which none of them are used in Arabic writing and vice versa.
- There are different writing styles for common word between Farsi and Arabic languages which cause the general shape of a word differs in two languages. Therefore segmentation, features extraction and also recognition techniques for a word in one of these languages are not suitable essentially in other language.
- …

Finally, it is necessary that more researches will be carried to provide better techniques to deal with Farsi OCR systems.

Appendix II

Character Segmentation

1. Introduction

In both online and offline OCR systems (including FOCR), there are two main approaches for automatic understanding of the cursive script - segmentation free (holistic, global, top-down) method (Ebrahimi & Kabir, 2008; El-Hajj, Likeforman & Mokbel, 2005; Bahmani, Alamdar, Azmi & Haratizadeh, 2010), and segmentation-based (analytical, bottom-up) method (Pirsiavash, Mehran & Razzazi, 2005; Zand, Naghsh Nilchi & Monadjemi, 2008).

In the first approach, each sub-word is treated as a whole and the recognition system does not consider a word as a combination of separate letters. The use of the holistic method has been inspired by researches in psychology, and it deals with image like human reading and vision. By consideration this fact that the segmentation process is considered as the main source of recognition errors, some researchers try to avoid this stage by using holistic recognition approach. In this approach, each sub-word is considered as a new class. Hence, holistic approach is usually used to recognize a limited set of words such as the name of cities and countries on mailing envelops, and legal and courtesy amounts in bank cheques processing (Noaparast & Broumandnia, 2009). In most all significant results of holistic methods, Hidden Markov Model (HMM) has been used as the recognition engine and some features such as total word length, different parts of a word in upper area or lower area of the baseline, ascenders, and descenders are used as features. The main drawbacks of this approach include: need to a large number of samples for each word to cover all available states; and limitation on the number of the classes (words) in a predefined lexicon (Dehghan, Faez, Ahmadi & Shridhar, 2000). These characteristic causes this kind of approach is limited to only static dictionary applications like bank cheque processing. Also, the main advantage of the holistic strategy is that it avoids very complex and difficult segmentation operation.

The second strategy, which has done extensively by researchers in OCR systems, segments each word into smaller classifiable units such as characters, pseudo characters, graphemes or slices, as the building blocks of a sub-word (Safabakhsh & Adibi, 2005; Nasrudin & Omar, 2008). This approach is similar to human writing skill. Projection analysis, connected component processing, contour tracing, and white space are some of the used common techniques in this category. Usually, many errors are generated in finding the exact locations of the segmentation points in this approach. As a result, the best reported result for Farsi words segmentation is 90.26% only for a limited words set (Alaei, Nagabhushan & Pal, 2010b), and this is not satisfactory result in a fully implemented OCR system. The main advantage of the analytical approach, however, is that it can handle lexicons of unlimited size (Al-Hajj, Likforman & Mokbel, 2009).

In the analytical approach, the segmentation process is used in two different ways - explicit and implicit. If the segmentation process is carried out completely and the characters are then recognized, the segmentation process is called explicit segmentation. If the segmentation and recognition operations are carried simultaneously to produce the overall results, the process is called implicit segmentation (hybrid, segmentation-by-recognition, recognition-based segmentation) (Pirsiavash, Mehran & Razzazi, 2005). In this case, a feedback loop links the output of the classification stage to the input of the character fragments combination stage, (Xiu, Peng, Ding & Wang, 2006).

In both cases holistic and analytical, it is a common practice to use a dictionary in post processing stage in order to increase recognition accuracy. Generally, comparing methods of analytical and recognition based segmentation approaches shows that the analytical methods provide greater interactivity, saving of computation, and simplifies the task of recognizer (Elnagar & Bentrcia, 2012).

Using another terminology in the text recognition researches, the word “segmentation” is used in two different ways: external (high level) segmentation, and internal (low level) segmentation (Nasrudin, Omar, Zakaria & Yeun, 2008). External segmentation (document structure analysis, page decomposition) involves separating the different blocks of a page into different categories such as tables, figures, curves, and also texts (Shirali, Manzuri & Shirali, 2007). Internal segmentation (decompose a word to letters) involves slicing a word from a text into some (pseudo) characters which make up that word (Jumari & A.Ali, 2002).

Farsi is basically a cursive scripts language, even in the machine-printed forms. All Farsi letters stick to another letter from the right side, and 25 out of 32 Farsi letters stick together from the left and right sides (Table 2.1) to make sub-words. This characteristic poses challenging problems in FOCR applications which use analytical approach recognition. However, segmentation is the bottleneck, crucial task and difficult stage in development of a FOCR system, because incorrectly segmented characters produce misclassification or rejection of characters during the recognition process. Any error in segmenting the basic shape of characters will produce a different representation of the character components. Its difficulty stems from the high variability of writing scripts and styles which could be affected by writers. Hence, it is considered the main source of recognition errors (Nawaz, Sarfaraz, Zidouri & Al-khatib, 2003; Mostafa, 2004).

Although some designed algorithms for cursive Latin words segmentation might carry over to Farsi handwriting, they are generally not adequate for this task (Jumari & A.Ali, 2002). In spite of the intense effort concerned with character recognition and the success realized by some researchers in limited area, no developed system has reached ideal solutions for Farsi cursive script segmentation problem. The special characteristics of Farsi language have caused segmentation operation in Farsi documents be more challenging than other languages. This subject causes available segmentation techniques for other languages are not completely suitable for Farsi and the performance of segmentation block in FOCR systems will be less than the Latin. Finally, cursive writing like Farsi, where the segmentation problem of text into distinguishable characters is the source of many errors for OCR applications, is still open area and not yet fully explored.

2. Printed Farsi/Arabic Words Segmentation

There are some well-known internal segmentation techniques for the printed Farsi/Arabic texts such as:

- Segmentation based on vertical projection profiles of the image (Sarfaraz, Nawaz & Al-Khuraidly, 2003; Zheng, Hassin & Tang, 2004).
- Segmentation based on contour tracing (Omidyegane, Nayebi & Azmi, 2005, 2007; Mehran, Pirsiavash & Razzazi, 2005; Azmi & Kabir, 2001).
- Segmentation based on thinning and skeleton tracing (Shaikh, Ali Mallah & Shaikh, 2009).
- Segmentation based on template matching (Abdullah, Al-Harighy & Al-Fraidi, 2012).

- Segmentation based on neural networks (Menhaj & Adab, 2002; Hamid & Haraty, 2001; Mehran, Pirsiavash and Razzazi, 2005).
- Segmentation based on morphological techniques (Abandah, Jamour and Qaralleh, 2014).
- Segmentation based on transforms (Broumandnia, Shanbehzadeh & Nourani, 2007).
- Segmentation based on hidden Markov models (Gouda & Rashwan, 2004).

However, inconsistency in shape of Farsi letters, variable gaps between words, different forms and different locations for secondary parts such as dots of each letter in handwritten Farsi words, have caused the existing segmentation methods for printed Farsi texts to be not completely applicable to handwritten Farsi documents (Rajabi, Nematbakhsh & Monadjemi, 2012). For example, the space between the sub-words in printed texts is normally one-third of the space between the words, but it varies in handwritten texts, or there is no overlapping between characters in printed texts, but it is frequently seen in handwritten documents (Lorigo & Govindaraju, 2006). These features lead to difficulties in internal segmentation for handwritten texts.

3. Handwritten Farsi/Arabic Words Segmentation

Farsi alphabet has been derived from Arabic alphabet, but different handwriting styles as well as some differences between Farsi and Arabic letters cause that the methods for segmentation Arabic words are not completely applicable for Farsi handwritten documents. However, Farsi has four more letters ‘گ’, ‘چ’, ‘پ’, and ‘ژ’ in its alphabet, compared to Arabic alphabet. Hence, it is a superset of Arabic alphabet, and the proposed methods for segmentation of the Farsi texts are applicable for the Arabic texts, too.

In 1996, Olivier, Miled, Romeo and Lecourtier decomposed a word to smaller units, called grapheme. A grapheme is usually a small portion of a single character. Then, they analyzed the generated graphemes using some heuristic rules to produce a code for each character. They tested their proposed method on 6000 words from IFN/ENIT dataset and achieved to 97.41% correct word segmentation to graphemes. However, the main drawbacks of grapheme-base segmentation systems are the large diversity of grapheme shapes, and necessity to post processing operation to combine graphemes to rebuild initial characters. In another effort, they proposed another method for segmenting handwritten words to graphemes and used HMM to handle this operation (Olivier, Miled, Cheriet & Lecoutier, 1997). They defined a large set of graphemes to increase the correct recognition rate. Using 232 handwritten words from IFN/ENIT dataset, they achieved 82.5% correct recognition rate in average.

Motawa, Amin and Sabourin (1997) employed morphological operators opening and closing to find regularities and singularities in contour of a word. Singularities include the vertical or semi-vertical strokes which might represent feature points such as start, end, branch, cross points, or a transition to another letter. Regularities are all the contour pixels without singularities and are used to connect two successive letters in a word. Regularities hold the information that is necessary for linking a letter to its subsequent letter. Hence, regularity points are the candidate for segmentation. Testing on a few hundred words, they achieved to 81.88% correct segmentation rate.

Hamid and Harati (2001) extracted 52 features for any column of a word image. They then identified the exact location of segmentation points in each word, manually. This information was fed into a feed forward neural network with 52 inputs, four hidden layers, and one output to verify the validity of these segmentation points. After using rejection strategy, their system achieved to 69.72% correct segmentation. However, their method could not handle segmentation for vertical overlapping of characters.

Sari, Souici and Sellami (2002) extracted candidate segmentation points from local minima in lower contour of an image using some heuristic rules. Their approach divides a sub-word into three upper, middle, and lower zones by finding baseline via horizontal projection histogram. Using two topological filtering acceptation and rejection rules, and for 100 handwritten words, 86.0% correct segmentation rate has been reported.

Lorigo and Govindaraju (2005) proposed an explicit over-segmentation method for handwritten Arabic texts. In the first stage, they measured stroke width, removed noise, and explored dots and sub-words. Then, they detected loops, optimized the baselines, and employed vertical and horizontal gradients in the baseline strip to find a large number of candidate segmentation points. This method can find segmentation points between two adjacent letters that those letters are too close together. Finally, they used some knowledge concerned to Arabic letter shapes to remove breakpoints in loops, at edges, and so on. They achieved to 92.3% correct segmentation using the images of 200 words from IFN/ENIT dataset.

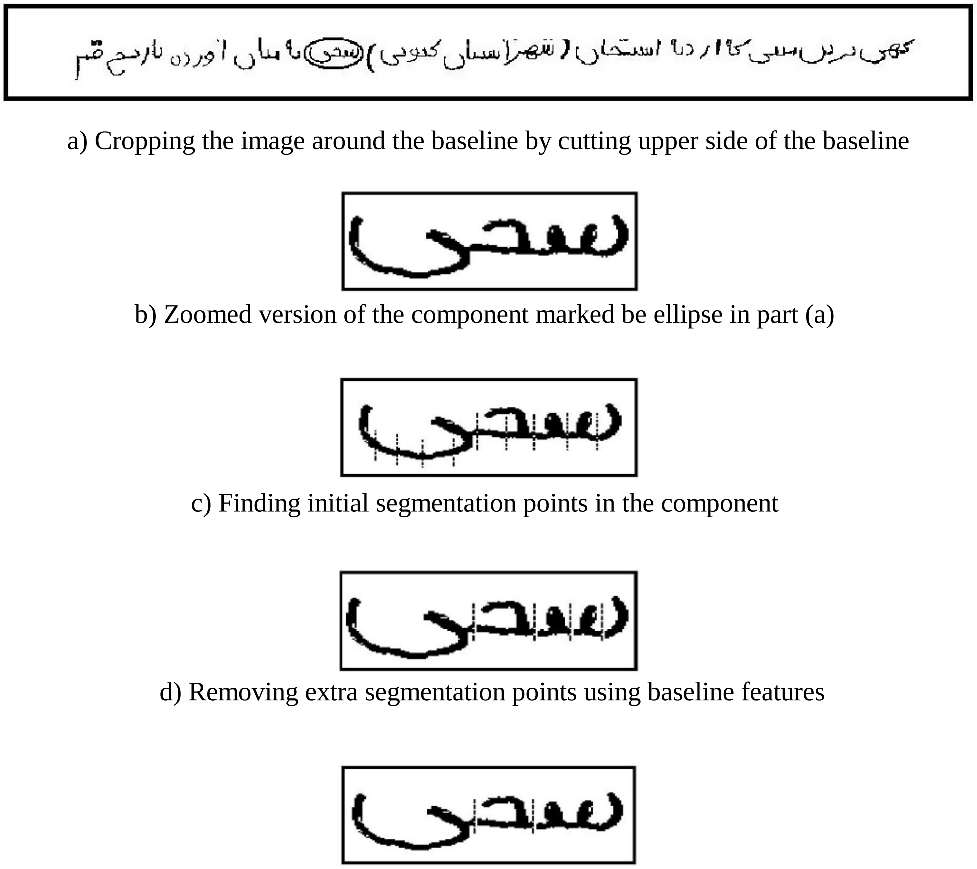
Safabakhsh and Adibi (2005) introduced a system for recognition of Farsi Nasta'aligh handwritten words using HMM. In segmentation phase, their system segmented words into pseudo characters using two methods, finding local minima of upper contour, and extracting regularities and singularities parts. Regularities and singularities are extracted from the contour of the words. They hold the information that is needed for linking a letter to its subsequent letter. Hence, regularity points are the candidate points for segmentation. These two groups of features are very suitable for internal segmentation of cursive texts.

Using local minima of upper contour, they achieved 95.68% correct segmentation on special private dataset TST1 of Nasta'aligh words.

In 2009, Wshah, Shi and Govindaraju tried to design a segmentation algorithm to create a small lexicon of handwritten Arabic sub-word including one grapheme, one letter, two letters, or three letters combination. Their algorithm is based on this fact that every Arabic connected letters have intersection points in their skeleton at the beginning and end of the letters. The mentioned method includes preprocessing, chain code generation, skeletonization, and finding exterior contour. After that, they build a distance map for each intersection point with all chain code members. By finding the smallest three paths from the chin code contour points to the skeleton, they find the final segmentation points. Their approach was achieved to 90.4% correct segmentation using 6300 words from DARPA dataset.

Elzobi, Al-Kamdi, Dinges and Michaelis (2010) first skeletonized the input text images. They then found a set of critical features points such as end, branch, loop, and dot points of the image contour. All columns in the thinned image were considered as potential segmentation points. Using some structural rules related to Arabic scripts, most of non-desirable segment points were removed. This technique was applied on a dataset with only 200 words and got a good accuracy in correct segmentation. However, most of overlapping and some of loops are segmented wrongly.

Alaei, Nagabhushan and Pal (2010b) proposed a baseline-dependent approach for segmentation of handwritten Farsi words. In the first step, they traced and straightened the baseline by utilizing a painting technique (Nagabhushan & Alaei, 2010). Using the vertical projection technique, they segmented a line into sub-words. At this stage, their method suggests a large number of candidate points for segmentation. To ignore wrong candidate segmentation points, they used some language-dependent rules that correspond to the Persian handwriting style and heuristic rules to overcome the extra segmentation. They achieved 93.49% correct segmentation for 200 handwritten Arabic words from the IFN/ENIT dataset. Figure 1 shows a sample of Alaei's proposed segmentation method.



a) Cropping the image around the baseline by cutting upper side of the baseline

b) Zoomed version of the component marked be ellipse in part (a)

c) Finding initial segmentation points in the component

d) Removing extra segmentation points using baseline features

Figure 1 : Word segmentation for handwritten Farsi/Arabic word (Alaei, Nagabhushan & Pal, 2010b)

Another effort for segmenting handwritten Arabic words to characters was carried out by Rasheed by using NN (Rasheed, 2011). After some common pre-processing operations such as noise removal and slant correction, he found the valleys in arcs between letters, which are ideal segmentation points by a heuristic method. Thereafter, a NN was trained with valid segmentation points from a dataset, and correctness of the segmentation points was assessed. In order to prevent wrong segmentation operation for letters such as 'ص' and 'م', the algorithm searched for existence of a hole in the main body of a letter. However, he did not report about correct segmentation rate.

Elangar and Bentrcia (2012) defined a multi agent approach for Arabic handwritten word segmentation in Naskh style. Based on this fact that most of the Arabic characters have the loop and cavity in their bodies, they employed seven agents: baseline, loop, letter seen, under baseline cavity, above baseline right cavity, above baseline left cavity, and above baseline narrow left cavity, to find initial candidate segmentation points from the word skeleton. Then, by using some topology-based rules, the validity of the candidate points is verified. For 550 selected words from IFN/ENIT dataset, 86.0% correct segmentation words to letters were obtained.

Some of the word segmentation methods for printed and handwritten Farsi texts are summarized in Table 1. It is clear that the methods which are based on structural and statistical features provide better results than others. However, it is not possible to compare the developed segmentation algorithms with each other, because they have not been tested on the same datasets, and most of them did not provide the results of the segmentation module. Finally, it is clear that no perfect and error-free segmentation technique is available yet and this area of research is still open for further enhancement.

Table 1 : Some Farsi words segmentation researches

	Year	Researchers	Segmentation method	Text type	Dataset size (words)	Accuracy %
1	1981	Parhami & Taraghi	Vertical projection	Printed	----	100
2	1995	Hashemi et al.	Contour Tracing	Printed	----	99.7
3	1996	Timsari & Fahimi	Morphological operators	Printed	2000	98.30
4	2001	Azmi & Kabir	Upper contour	Printed	8056 Sub-word	98.5
5	2002	Menhaj & Adab	Recognition-based Segmentation	Printed	Only 3 fonts of size 72	90.00
6	2005	Safabakhsh & Adibi	local minima of upper contour	Handwritten Nasta'aligh	50-Words dictionary	95.68
7	2005	Mehran et al.	Vertical Projection, Contour Tracing, distance from baseline	Printed	40,000 sub-word	98.7
8	2007	Omidyeganeh et al.	Vertical Projection, Contour Tracing	Printed	----	99.64
9	2007	Broumandnia & Shanbehzadeh	Wavelet transform	Printed	1000	97.83
10	2008	Zand et al.	Recognition-based Segmentation	Printed	----	----
11	2010	Alaei et al.	Vertical Projection, Language dependent rules	Handwritten	200	93.49

4. Summary

In comparison, the holistic method usually outperforms the analytical approach, but it still needs a more detailed model of the language, which its complexity grows as the vocabulary set gets larger. In addition, in the first method, the number of recognition classes is far more than classes in segmentation based methods. Hence, this method seems to be not feasible duo to the numerous numbers of words in a language. However, if the second case is used, existing results show the segmentation of a cursive word is a very difficult problem. The main advantage of the analytical approach, however, is that it can be used to recognize infinitely large vocabularies.

The results obtained from the use of the analytical approach show that the accurate segmentation of a handwritten cursive word to letters is very difficult. Different segmentation methods may produce different (pseudo) characters, and as a result, it degrades the classifier performance. Over-segmentation is the main drawback of the most of the current segmentation techniques. It happens because the majority of available segmentation algorithms are tuned to find all candidate segmentation points in a sub-word. Also, there are some simple shapes in the body of some Farsi letters that resemble other letters. Hence, it is important to correct these types of errors at the post-processing stage.

In both the holistic and the analytical methods, it is a common practice to use a dictionary in the post-processing stage in order to increase recognition accuracy. Generally, a comparison between the analytical and the recognition-based segmentation approaches shows that the analytical methods provide greater interactivity, require less computation, and simplify the task of the recognizer (Elnagar & Bentrica, 2012). Usually, five methods have been principally used for Farsi letter segmentation:

4.1 Curvature analysis and feature points detection using thinning operation: Some segmentation methods use the thinning operation as a first step. They will then look for some feature points such as: branch, intersection, start and end points in the image skeleton. Using these feature points, a sub-word is broken into a number of graphemes. By using a set of predefined rules, the graphemes are then joined to make the initial letters. However, different thinning algorithms may produce different thinned images. Moreover, the thinning process might alter the shape of a letter, especially the poor quality letters. It is the main drawback of this type of segmentation methods.

4.2 Outer contour analysis: This technique is used to find the local minima of the upper contour of an image. These local minima points are the initial segmentation candidate points which are then filtered to find the final decisive segmentation points, using some language-based rules (Azmi & Kabir, 2001). The contour-tracing method avoids all problems which had resulted from the thinning process by analyzing the structural shape and morphological features of characters as they have been scanned. In many cases, however, the contour must first be smoothed.

4.3 Vertical projection and stroke thickness calculation: In this approach, the local minima in the vertical projection of each sub-word indicate the candidate segmentation points (Parhami & Taraghi, 1981; Mehran, Pirsiavash and Razzazi, 2005; Omidyeganeh, Azmi, Nayebi & Javadtalab, 2007; Alaei, Nagabhushan & Pal, 2010b). However, this method is completely dependent on finding the correct location of the baseline. Hence, if the correct baseline is not found, the segmentation block will not function as expected. This technique works very well for printed characters which do not overlap each other. The method produces very poor results when ligatures and overlapped characters are present. The main advantage of this method, however, is that it is not dependent on the shape, size, and font of the characters.

4.4 Singularities and regularities: These two features are extracted from the contour of a word. Singularities include the vertical or semi-vertical strokes which might represent feature points such as start, end, branch, cross points, or a transition to another letter.

Regularities are all the contour pixels without singularities and are used to connect two successive letters in a word. They hold the information that is needed for linking a letter to its subsequent letter. Hence, regularity points are the candidate for segmentation. It seems these two groups of features are very suitable for cursive text internal segmentation.

4.5 Recognition-based Segmentation: This technique separates a narrow strip from the input image, and tries to recognize it. If the recognition operation fails, it will then increase the width of the strip by one or more pixel(s) and will try to recognize it again. This process is repeated until the last letter in a sub-word is recognized (Menhaj & Adab, 2002; Zand, Naghsh Nilchi & Monadjemi, 2008). The main advantages of the recognition-based segmentation technique are that it bypasses serious letter separation problems, and, it is also not necessary to identify the accurate character segmentation path.

Appendix III

Principal Component Analysis

Principal Component Analysis (PCA) is a statistical representation of random variable to find important patterns in high-dimension input data. It is a pre-processing step in recognition applications which involves converting a correlated features space to a new non-correlated features space. In the new space, features are reordered in decreasing variance value such that the first transformed feature accounts for the most variability in the data. Hence, PCA overcome the problem of high-dimensionality (Wang, 2011).

For employing PCA as a tool in PR applications, it is necessary to carry three important steps including: generation transformation matrix by using training data; projection of data input into transformation matrix; and finally projection test data into generated new space. Basically, this process is started by converting a 2-D image A_i from training dataset to 1-D image by concatenating the rows of data image consequently. This new vector is a k×m elements vector which parameters k and m are the dimension of the initial image A_i. Then, by stacking all 1-D vectors A_i, the training matrix A, which has (n) × (k×m) dimensions, is produced. Parameter n is the total number of all training patterns in all classes in the pattern space. In training matrix A, all similar input class images should put after each other. Hence, each column of training matrix A is indicating one feature (dimension) of training samples. Consider X as a random vector population, where:

$$X = (x_1 , \dots , x_n)^T \tag{1}$$

The mean of each dimension is calculated:

$$\mu_x = E\{x\} \tag{2}$$

In order to normalize data values in each dimension, data are reduced from corresponding mean. It causes that mean of each dimension is changed to zero (Equation 3).

$$A = [A_i]_{i=1}^{n} = [x_i - \mu_i] \tag{3}$$

In Equation 3, μ_i is the mean of class i and $x_i - \mu_i$ (i=1,…,n) are the columns of matrix *A*. Next step is computing covariance matrix *C* as follows:

$$C = \frac{1}{M}\sum_{i=1}^{M} A_i * A_i^T \tag{4}$$

$$C_x = E\ \{ (x-\mu_x)\ (x-\mu_x)^T \} \tag{5}$$

The components of C_x, represent the covariance between the random variable components x_i and x_j. If two components x_i and x_j are uncorrelated, their covariance is zero. By solving the characteristic equation:

$$|Cx - \lambda I\ | = 0 \tag{6}$$

where *I* is the identity matrix, λ coefficients could be found. If *A* is a matrix consisting of eigenvectors of the covariance matrix as the row vectors, by transforming a data vector *X*, we get:

$$y = A\ (X-\mu_x\ ) \tag{7}$$

which is a point in the orthogonal coordinate system defined by the eigenvectors. Components of *y* can be seen as the coordinates in the orthogonal base. The original data vector *X* is reconstructed from *y* by:

$$X = A^T\ y + \mu_x \tag{8}$$

The original vector was then reconstructed by a linear combination of the orthogonal basis vectors. This means that we project the original data vector on the coordinate axes having the dimension *K* and transforming the vector back by a linear combination of the basis vectors. This minimizes the mean-square error between the data and this representation with given number of eigenvectors. If the data is concentrated in a linear subspace, this provides a way to compress data without losing much information.

The covariance matrix *C* is a square matrix and therefore, the eigenvalues and eigenvectors of it can be calculated. The eigenvectors are generally perpendicular to each other. The eigenvector with the biggest corresponding eigenvalue is the most significant representative of data and it is considered as the first most significant principal component. The eigenvector with the second biggest corresponding eigenvalue is considered as the second most significant principal component and so on. Therefore, by sorting the eigenvalues by the eigenvalues in descending order, we can find the most eigenvectors as most significant representation data. By picking the eigenvectors having the largest eigenvalues, a little information is lost in the mean-square sense (Bouchareb, Hamidi & Bedda, 2008). New feature vector will be made using the *k* first significant eigenvectors as Equation 9. In this equation ev_i is i'th most significant eigenvector.

$$FV = (ev_1\ ev_2\ \dots\ ev_k) \tag{9}$$

Feature vector *FV* contains the features ordered column by column. The first column contains the most important feature vector, and the last column contains the least important feature vector of initial data. By multiplying derived *FV* in the initial training dataset *A*, the final training dataset *B* solely in terms of selected eigenvectors will be created as follows:

$$B = FV \times A \tag{10}$$

Appendix IV

Various Distance Measurement Functions

In some PR applications such as OCR, the conventional similarity measure is defined as follows:

$$S\ (X,M) = \frac{<X\ ,M>}{\|X\|\ .\ \|M\|} \tag{1}$$

which X and M are the input data and mean vector (or template) of a class respectively. The notation < , > denotes the inner product and the notation ‖ ‖ denotes the norm of a class.

Suen et al. increment one unit to a variable named similarity variable, per two corresponding pixels of the image (f(x,y)) and related template T_i(x,y) for class C_i , if the values of both pixels are '1' (Eq. 2). In this approach, similarity variable will have a value in range of 0 to n×m (n and m are image dimensions) [Suen et al., 1977].

$$S_i\ (\ f\ ,\ T_i\ ) = \sum_{x=1}^{n} \sum_{y=1}^{m}\ [\ f(x,y)\ \ AND\ \ T_i(x,y)\ ] \tag{2}$$

f (x,y) : Binary input image

T_i (x,y) : Binarized Template Matrix (BTM)

There are other definitions for similarity measurement and other methods for comparing a sample with the calculated template for a special class and computing SiV for that sample.

Like to Suen et al., Sitamahalakshmi et al. used following different forms of similarity function to compare an instance input with prototypes of different classes. They used Hamming distance, linear correlation, and cross correlation values (Eq. 3 to 5 in order), and combined them using Dempster-Shafer theory to achieve a better match in a typical character recognition system [Sitamahalakshmi et al., 2010].

$$\text{Hamming Distance:} \quad H\,(f\,,T_i) = \sum_{x=1}^{n} \sum_{y=1}^{m} [\, f(x,y) \;\; XOR \;\; T_i(x,y)\,] \qquad (3)$$

$$\text{Linear Correlation:} \quad LC\,(\,f\,,T_i\,) = 2 \times \frac{S_i\,(f\;,\;T_i\,)}{(\,N_f + N_{T_i}\,)} \qquad (4)$$

$$\text{Cross Correlation:} \quad CC\,(\,f\,,T_i\,) = \frac{{S_i}^2(f\;,\;T_i\,)}{(\,N_f * N_{T_i}\,)} \qquad (5)$$

In mentioned equations, f and T_i are input image and related class template, and N_f and N_{Ti} are densities of them respectively.

Sawaki and Hagita used complementary similarity measurement for document headlines recognition as follows [Sawaki and Hagita, 1998]:

$$S_C\,(\,F,T\,) = \frac{a.e-b.c}{\sqrt{T.(n-T)}} \qquad (6)$$

where:

$a = \sum_{i=1}^{n} f_i \,.\, t_i$, $b = \sum_{i=1}^{n} \,(1 - \,f_i\,)\,.\, t_i$

$c = \sum_{i=1}^{n} f_i \,.\, (\,1 - \,t_i\,)$, $e = \sum_{i=1}^{n} \,(1 - \,f_i\,)\,.\,(\,1 - \,t_i)$

$T = \,\|\, T \,\|$, $a+b+c+e = n$

Downton et al. used cosine function (as below) for representing similarity level between sample c and template t [Downton et al., 1988]:

$$S_i\,(c,t) = \frac{c^T\,t}{|c|\,.|t|} \qquad (7)$$

c : vector of a sample

t : vector of a template correspond to class sample

|c| and |t| : magnitude of c and t respectively

Equation 7 is converted to equation 8 in binary images simply:

$$S_i^2\,(c,t) = \frac{n_m^2}{\mathrm{n}_c\,.\,\mathrm{n}_t} \qquad (8)$$

n_m : the number of '1' pixels which intersect between c and t

n_c : the number of '1' pixels in c

n_t : the number of '1' pixels in t

Sadri et al. took a modified version of Rogers_Tanimoto similarity measure S_{R_T} (X,Y) to detect the similarity value between a new binary image (features) X and binary template of a cluster Y in order to determine the optimal number of clusters in the input data [Sadri et al., 2006]. Equations 9 and 10 represent in order Rogers_Tanimoto similarity measure S_{R_T} (X,Y) and modified version of it, proposed by Sadri et al.

$$S_{R\_T}(X,Y) = \frac{X^t\,Y + \bar{X}^t\,\bar{Y}}{X^tY + \bar{X}^t\,\bar{Y} + 2X^t\bar{Y} + 2\bar{X}^t\,Y} \qquad (9)$$

$$MS_{R\_T}(X,Y) = \frac{\alpha X^t\,Y + \beta\,\bar{X}^t\,\bar{Y}}{\alpha X^tY + \beta\,\bar{X}^t\,\bar{Y} + 2X^t\bar{Y} + 2\bar{X}^t\,Y} \qquad (10)$$

$$\alpha\ \geq 1 \quad , \quad \beta\ \geq 0$$

In equation 10, coefficients α and β are two adjustable weights, and X^t Y represents inner product of X and Y. Other definitions for similarity measures and distances such as

Minkowski distance (L_P distance), Bottleneck distance, Hausdorff distance, Turning Function distance, Frechet distance, Reflection distance and Transport distance (Earth Mover's distance) can be found in literature [Veltkamp, 2001].

Appendix V

Random Projection

Random Projection (RP) is one of the most-used methods for dimensionality reduction in PR domain (Fodor, 2002). Time complexity of this method is *O*(npq) (Frad2003) which n is number of sample points, p is the dimension of initial space, and q is the dimension of the reduced space. Lower time complexity of RP technique compared to other dimensionality reduction methods such as PCA ($O(p^2n)$ + $O(p^3)$), has caused that the researchers employ RP in several applications.

RP is based on matrix manipulation, which uses a random matrix to project the original data set into a lower dimensional subspace (Deegalla & Bostrom, 2006). A simple description of RP technique states that for a set of *n* sample points in initial *p*-dimensional Euclidean space there exists a linear transformation of data points into a secondary *q*-dimensional space, where $q \geq O(\epsilon^{-2} log(n))$ that preserve distances up to a factor 1 ± ϵ. Another representation for bounds on ϵ and q has been defined by Dasgupta and Gupta (1999) as follows:

$$q \geq 4 * \left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \ln(n) \qquad (1)$$

Achlioptas (2001) added probability into the random projections issue and introduced a simpler way for introducing it. He considered n samples with initial p features as initial matrix X (n × p matrix). Also, he chose ϵ, β, and $q \geq \frac{4+2*\beta}{\frac{\epsilon^2}{2}-\frac{\epsilon^3}{3}} \ln(n)$, and let $E = \frac{1}{\sqrt{q}}XP$, that P is projection matrix. Now, the mapping from X to E preserves distance up to factor 1 ± ϵ

for all rows in X with probability ($1 - n^{-\beta}$). R_{p*q} , random projection matrix, is constructed in one of the following ways:

- $r_{ij} = \pm 1$ with probability 0.5 each.
- $r_{ij} = \sqrt{3} \times (\pm 1$ with probability 1/6 each, or 0 with probability 2/3).

Appendix VI

Partitioning Method for Dataset size Reduction

1. Introduction

The main idea behind the dataset size reduction operation is that in a large dataset, some of the training samples in a class might fully or nearly resemble each other, and therefore, they cannot produce different crucial information to the training part of a PR system. Hence, if the (almost) similar samples in a class are found, then it is possible to keep only some of them and remove the rest from dataset. As a result, the volume of a dataset can be decreased.

In the most of available methods for dataset size reduction, researchers compare all samples from a class to all samples in another class to find the boundary points between all class pairs. Finally, they put these samples in final dataset as important key samples (or delete them as unimportant samples in making system model). It is obvious that this approach need to compare all classes' samples with other classes' samples and therefore it is very time consuming process and also it needs to very huge memories for storing of samples' characteristics. The proposed Partitioning Approach (PA) uses a completely different approach in compare to the other available methods. Here, the samples of a class are compared to other samples in the same class, based on their similarities to that class template.

The proposed method PA uses Modified Frequency Diagram (MFD) matching technique (Section 4.4.2; Equation 4.3) to create a template for each class. In order to reduce processing time, the proposed method compares the samples of a class to other samples in the same class, instead of comparing samples from different classes. The similarity measurement function (Equation 4.5) is then used to compute a similarity value for each training sample of a dataset corresponding to that class template (Section 4.4.4). Then, the similarity values are normalized into closed interval [0, 1]. Thereafter, the samples in a specific class are sorted in descending order and are rearranged based on the calculated similarity values. The reordered samples of each class are split into two separate partitions. The first partition includes the most similar samples to the class template, and the second partition includes the rest of the samples of that class. One sample of the first partition and all samples of the second partition, for each class, are kept in the final reduced dataset version. This approach keeps all the outlier samples in each class, which are important for evaluating the system efficiency (because outlier samples are not usually similar to class template), and adjusting the recognizer parameters.

The proposed algorithm has four main parts including: 1) Template generation for each class (Section 4.4.1); 2) Template binarization (Section 4.4.2); 3) Calculating similarity value (Section 4.4.3); and 4) Deleting some patterns from each class and reducing the class size. The initial input to this module is the output of pre-processing module (Section 4.3). The steps 1 to 3 of the proposed dataset size reduction method have been explained in Sections 4.4.1, 44.2, and 4.4.3, in order. The last step – reduction operation – is explained as follows.

2. Reduction Operation

For dataset size reduction, two concepts **S**imilarity **V**alue (SV) and **S**imilarity **I**nterval (SI) were used. The result of Equation 4.5 is a numeric vector from minimum value ***A*** to maximum value ***B*** for each class which they indicate the SV between the samples of a specific class with the corresponding class template. Using the normalization Equation 1, a SV is changed to **S**imilarity **R**atio (SR) in the closed interval [0,1].

$$\mathrm{SR} = (\mathrm{SV} - A) / (B - A) \tag{1}$$

In practice, through the computation process and also because of the nature of the samples in PR systems (especially in handwritten OCR datasets), the computed SR variables usually differ from each other a little. Hence, if the method wants to categorize all samples with an exact equal SR into one cluster, then only a few samples will be held in each cluster, and only a few samples will be removed from the dataset. To overcome this drawback, the new parameter SI was defined and used instead of SR parameter. For example, the SI [0.9 – 1] means creating a cluster of samples of a specific class, in which all of the samples have a SR from 0.9 to 1.

SI has a vital role in this approach. Selecting small SI causes only a few number of patterns (with similarity value very close together) are put in a category, and therefore, only a few number of patterns are removed from training dataset. In contrary with, selecting large SI causes a large number of patterns (which they are not very similar together) are put in a category. Therefore, a valuable decrease in dataset volume is occurred, but some of non-similar patterns, with valuable different information for system training, are removed from dataset, undesirable. Selecting the SI depends to different important factors such as: type of used system, the type of data, recognition accuracy, and so on. The computed SVs and SIs were used to delete some of the training dataset samples and decrease the training dataset size.

3. Partitioning Approach (PA)

In this approach, the available samples in each class were sorted in decreasing order, based on their SVs, and they were then divided into two separate partitions. Using SI [n - 1], in which n<1, the partition ‘1’ includes the most similar data to the class template, and the partition ‘2’ includes the rest of the samples of that class. Finally, one sample of partition ‘1’ and all samples of partition ‘2’ of each class were kept in the final Reduced Dataset (***R_Dataset)*** version. Figure 1 depicts the overall process for this method.

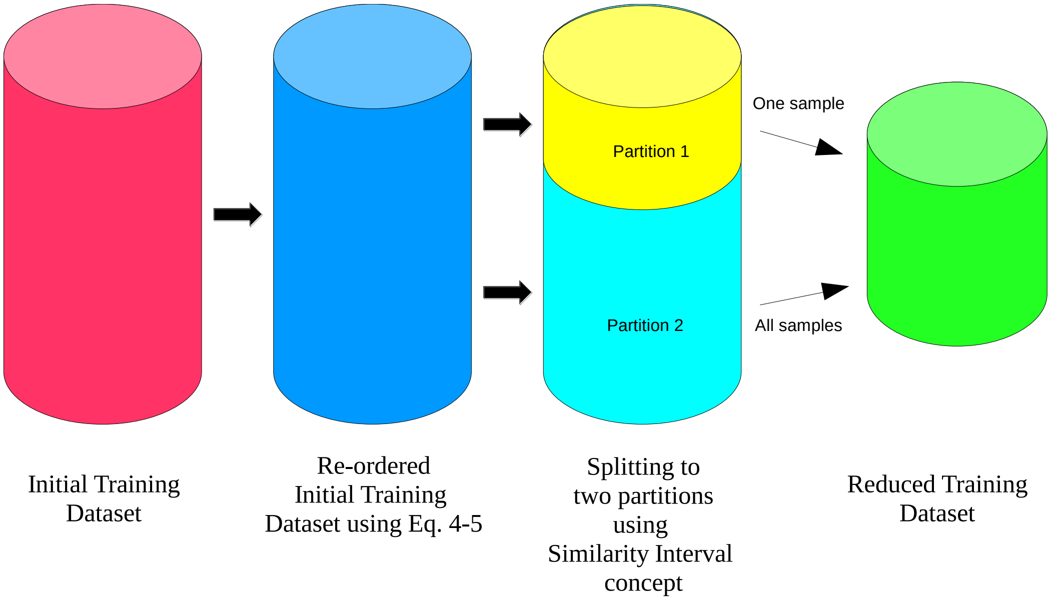

One sample
Partition 1
Partition 2
All samples
Initial Training Dataset
Re-ordered Initial Training Dataset using Eq. 4-5
Splitting to two partitions using Similarity Interval concept
Reduced Training Dataset

Figure 1 : The flow diagram for dataset size reduction - Partitioning approach

3. Experimental Results for the Proposed Dataset Size Reduction Method PA

In order to evaluate the efficiency of this dataset size reduction method and understanding that how much this method has negative effect on system accuracy in an FOCR system, it was applied on training digits and letters parts of the Hoda dataset. The original training part of the Hoda dataset has 60,000 samples for digits and 70,645 samples for letters.

Based on section 4.4, the available samples in each class were categorized into two partitions. Using SI [0.80 - 1], partition '1' included all samples that were 80% or more similar to a class template, and the rest of the samples were included in partition '2'. One sample of partition '1' and all samples of partition '2' were kept for each class to make a ***Reduced*** version of the training ***Dataset*** (***R_Dataset***). This procedure was repeated for similarity intervals [0.85 - 1], [0.90 - 1], and [0.95 - 1], too.

In order to compare the proposed model with other clustering methods, k-means clustering technique was applied on Hoda dataset to create two clusters of each available classes, based on their similarity values. Thereafter, similar to proposed dataset size reduction technique, one sample of cluster '1' and all samples of cluster '2' moved to final reduced version of digits and letters datasets. The results of reduction operations are reported in Table 1 (for digits) and in Table 3 (for letters).

3.1 Experiments on Digits Part of the Hoda Dataset

Columns 3 to 6 of Table 1 show the number of samples in each class of digits part of the Hoda dataset, after applying the PA method on training dataset for the mentioned similarity intervals. Using PA method, the initial dataset volume was reduced from 100% to 30.03%, 42.88%, 60.84%, and 85.12% for each selected similarity interval, respectively. By using Equation 2, the reduction rates for the digits part of the Hoda dataset for similarity intervals [0.80 - 1], [0.85 - 1], [0.90 - 1], and [0.95 - 1] were computed as 69.97%, 57.12%, 39.16% and 14.88% for each selected similarity intervals, respectively.

$$\text{Reduction rate} = \frac{N_i - M_i}{N_i} \times 100 \qquad (2)$$

N_i : No. of samples in dataset before removing similar samples

(ΣN_i = 60,000 for digits, ΣN_i = 70,645 for letters)

M_i : No. of samples in dataset after removing similar samples

The last column of Table 1 shows the number of samples in each class, using k-means (k=2) clustering technique. In this case, the initial dataset volume was reduced from 100% to 20.83% (i.e. reduction rate is 79.17%) with recognition time 0.0663526 second per sample.

Table 1 : The number of samples in digits part of the Hoda dataset after applying the proposed reduction algorithm, partitioning approach

Class Number (C_i)	**Farsi Digits Shapes for Class C_i**	**The number of samples (M_i) in each class after removing similar samples using different similarity interval**				**The number of samples (M_i) in each class after k-means clustering**
		[0.80 – 1]	**[0.85 – 1]**	**[0.90 – 1]**	**[0.95 – 1]**	
0	۰	130	219	460	1,261	1,357
1	۱	865	1,490	2,593	4,436	1,373
2	۲	1,752	2,653	3,860	5,332	1,250
3	۳	1,856	2,875	4,250	5,838	1,318
4	۴	1,778	2,443	3,586	5,892	1,758
5	۵	4,083	4,995	5,843	6,000	1,403
6	۶	1,718	2,425	3,534	5,808	1,582
7	۷	2,341	3,358	4,517	5,612	1,339
8	۸	2,184	3,103	4,353	5,571	749
9	۹	1,313	2,165	3,507	5,323	368
Total (ΣM_i)		**18,020**	**25,726**	**36,503**	**51,073**	**12,497**
Ratio of Reduced dataset Volume to Initial Dataset Volume $\frac{\Sigma Mi}{60{,}000} \times 100$		**30.03%**	**42.88%**	**60.84%**	**85.12%**	**20.83%**

Generally, in a PR system, the final performance depends heavily on the feature extraction phase. It is possible that a PR system with common methods in pre-processing and classification stages, by extracting a set of efficient features produces better results in compared to a PR system which uses powerful pre-processing techniques and employs a powerful classifier, but using a set of weak features, that they cannot represent patterns good enough. In order to nullify the effect of popular feature extraction methods, and to investigate only the effect of the proposed dataset size reduction algorithm, an FOCR system without prevalent feature extraction block was used. This causes the overall system performance to be dependent only to the other blocks such as pre-processing and classification blocks.

Similar operations were applied on input data in pre-processing stage for training and testing samples. Outputs of the pre-processing block were 50×50 pixels enhanced images. In order to speed up operations, the images size was changed to 20×20 pixels version, firstly. Then, the rows of each image were concatenated sequentially to make one 1D feature vector with length 400. In the classification block, the 400-dimensional feature vectors were directly fed to a *k*-NN classifier.

In the first experiment, the system was trained by using all initial 60,000 training samples, and then tested with all 20,000 testing samples of digits part of the Hoda dataset. Finally, the system achieved 96.49% recognition accuracy, with recognition time of 0.3787 second per sample. The experiment was repeated four times, using one of the ***R_Datasets*** versions with 18,020, 25,726, 36,503, and 51,073 training samples, corresponding to columns 3 to 6 of Table 1. The achieved recognition accuracies were 94.62%, 95.26%, 95.79%, and 96.18% for similarity intervals [0.8 – 1], [0.85 – 1], [0.9 – 1], and [0.95 – 1], respectively. The results are shown in Figure 2. In other experiment, the reduced version dataset created by k-means clustering (the last column of Table 1) was employed for training. Here, the accuracy significantly decreased from initial value 96.49% to 72.44%. This result was lower than the all achieved results in other experiments.

To investigate the superiority of the ***R_Dataset*** (generated by the proposed reduction method) against other subsets of the initial training datasets with the same size (i.e. $\sum M_i$), M_i samples (column 3 to 6 of Table 1) were selected randomly from each original class C_i before removing any samples. This operation produced another training dataset ***S*** with the same $\sum M_i$ samples. This operation was repeated four times, and in each of the iterations, a different training dataset **S_1, S_2, S_3,** or **S_4** was created randomly. The system was then trained with this newly generated **S_i** sets, and the trained system was used to recognize the Hoda testing digits (20,000 samples). The mentioned experiments were carried by using the **S_1, S_2, S_3,** and **S_4**, and the results have been shown in Figure 2. Also, Table 2 shows the recognition time for a test digit sample in experiments.

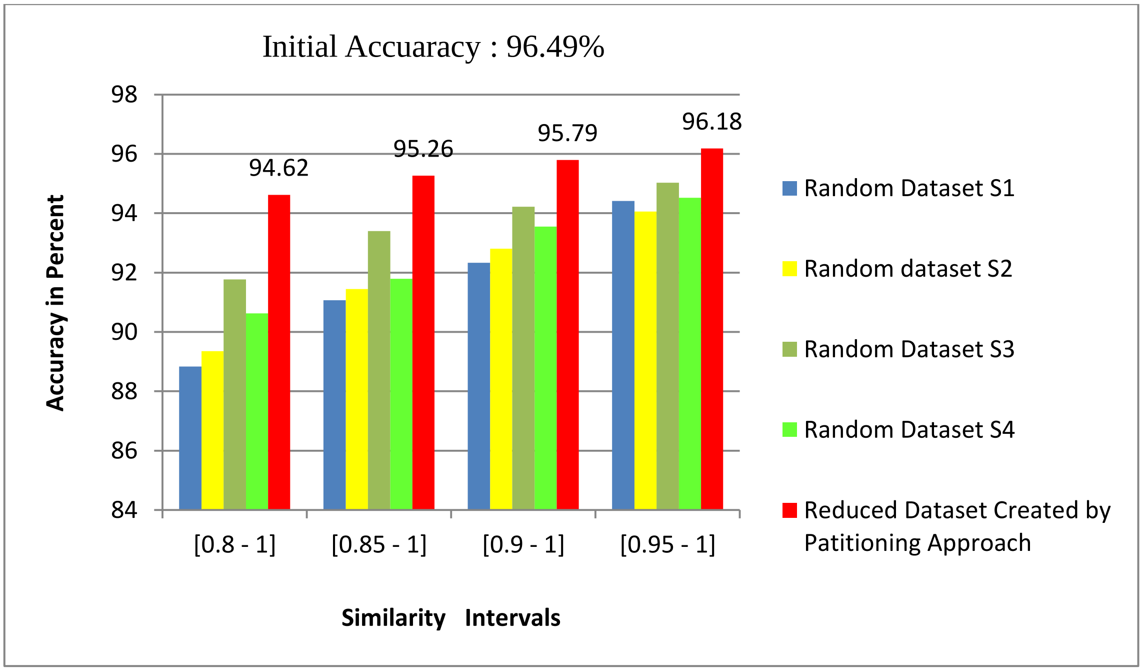


Figure 2 : Recognition results of a FOCR system (digits) using the proposed reduced dataset and also using random subsets **S_1** to **S_4**

Table 2 : Recognition time using different versions of reduced datasets (digits samples)

Similarity Interval for System Training	No. of Samples used for training the system using *k*-NN classifier	Recognition Time for a Sample (Seconds)	Ratio of Recognition Time to Initial Recognition Time T1
Using all training samples	60,000	T1 = 0.3787261	T1/T1 = 100%
[0.95 – 1]	51,073	T2 = 0.3303165	T2/T1 = 87.22%
[0.90 – 1]	36,503	T3 = 0.2367765	T3/T1 = 62.52%
[0.85 – 1]	25,726	T4 = 0.1688528	T4/T1 = 44.58%
[0.80 – 1]	18,020	T5 = 0.1233638	T5/T1 = 32.57%

3.2 Experiments on Characters Part of the Hoda Dataset

The aforementioned experiments in previous section for digits recognition were repeated again for characters part of the Hoda dataset. Columns 3 to 6 of Table 3 show the number of samples in each class of characters part of the Hoda dataset after the reduction operation for the mentioned similarity intervals. Using this method, the initial dataset volume was reduced from 100% to 48.54%, 65.08%, 79.32%, and 91.66% for each selected similarity interval, respectively. By using Equation 2, the reduction rates for the characters part of the Hoda dataset for similarity intervals [0.80 – 1], [0.85 - 1], [0.90 - 1], and [0.95 - 1] were computed as 51.46%, 34.92%, 20.68% and 8.34% for each selected similarity intervals, respectively.

The last column of Table 3 shows the number of samples in each class, using k-means (k=2) clustering technique. In this case, the initial dataset volume was reduced from 100% to 24.74% (i.e. reduction rate is 75.26%) with recognition time 0.0903289 second per sample.

Table 3 : The number of samples in characters part of the Hoda dataset after applying the proposed reduction algorithm, partitioning approach

Class Number (C_i)	**Farsi Letters Shapes for Class C_i**	**The number of samples (M_i) in each class after removing similar samples using different similarity interval**				**The number of samples (M_i) in each class after k-means clustering**
		[0.80 – 1]	**[0.85 – 1]**	**[0.90 – 1]**	**[0.95 – 1]**	
1	ا	652	982	1,467	1,726	438
2	ب	715	1,029	1,463	1,803	383
3	پ	1,073	1,544	1,934	2,010	423
4	ت	821	1,175	1,692	1,885	382
5	ث	1,378	1,657	1,747	1,803	557
6	ج	1,220	1,548	1,852	1,979	611
7	چ	1,173	1,494	1,755	1,883	588
8	ح	1,096	1,432	1,780	1,922	644
9	خ	897	1,287	1,711	1,917	539
10	د	951	1,269	1,633	1,880	589
11	ذ	435	708	1,144	1,528	401
12	ر	912	1,286	1,725	1,929	516
13	ز	721	1,046	1,588	1,873	355
14	ژ	1,265	1,766	1,873	1,975	474
15	س	491	811	1,318	1,736	295
16	ش	1,117	1,496	1,845	1,973	621
17	ص	431	718	1,234	1,631	414
18	ض	736	1,135	1,616	1,868	587
19	ط	1,134	1,549	1,940	2,001	548
20	ظ	1,263	1,624	1,750	1,856	473
21	ع	1,379	1,730	1,928	2,016	633
22	غ	1,018	1,414	1,839	1,991	491
23	ف	951	1,308	1,754	1,927	501
24	ق	1,033	1,400	1,769	1,914	585
25	ک	1,696	1,956	2,007	2,043	494
26	گ	1,294	1,681	1,838	1,962	518
27	ل	802	1,111	1,551	1,837	499
28	م	363	553	880	1,491	316
29	ن	1,431	1,775	1,922	2,007	568
30	و	1,007	1,402	1,757	1,955	530
31	ه	1,558	1,807	1,936	1,992	744
32	ی	1,395	1,690	1,905	1,981	615
33	ئـ	510	857	1,348	1,702	519
34	آ	524	592	623	636	158
35	هـ	579	808	1,240	1,672	334
36	ـه	272	338	413	451	133
ΣM_i		**34,293**	**45,978**	**56,613**	**64,755**	**17,476**
Ratio of Reduced dataset Volume to Initial Dataset Volume $\frac{\Sigma \text{Mi}}{70,645} \times 100$		**48.54%**	**65.08%**	**79.32%**	**91.66%**	**24.74%**

Here, in the first experiment, the system was trained by using all initial 70,645 training samples, and then tested with all 17,706 testing samples of characters part of the Hoda dataset. Finally, the system achieved 80.67% recognition accuracy, with recognition time of 0.4456 second per sample. The experiment was repeated four times, using one of the ***R_Datasets*** versions with 34,293, 45,978, 56,613, and 64,755 training samples, corresponding to columns 3 to 6 of Table 3. The achieved recognition accuracies were 64.13%, 72.48%, 78.47%, and 80.39% for similarity intervals [0.8 – 1], [0.85 – 1], [0.9 – 1], and [0.95 – 1], respectively. The results are shown in Figure 3.

In other experiment, the reduced version dataset created by k-means clustering (the last column of Table 3) was employed for training. Here, the accuracy meaningfully decreased from initial value 80.67% to 45.48%. This result was very lower than the all achieved results in other experiments.

To investigate the superiority of the ***R_Dataset*** (generated by the proposed reduction method) against other subsets of the initial training datasets with the same size (i.e. $\sum M_i$), M_i samples (column 3 to 6 of Table 3) were selected randomly from each original class C_i before removing any samples. This operation produced another training dataset ***S*** with the same $\sum M_i$ samples. This operation was repeated four times, and in each of the iterations, a different training dataset $\mathbf{S_1}$**,** $\mathbf{S_2}$**,** $\mathbf{S_3}$**,** or $\mathbf{S_4}$ was created randomly.

The system was then trained with this newly generated $\boldsymbol{S_i}$ sets, and the trained system was used to recognize the Hoda testing letters (17,706 samples). The mentioned experiments were carried by using the $\mathbf{S_1}$**,** $\mathbf{S_2}$**,** $\mathbf{S_3}$**,** and $\mathbf{S_4}$, and the results have been shown in Figure 3. Also, Table 4 shows the recognition time for a test letter sample in different experiments.

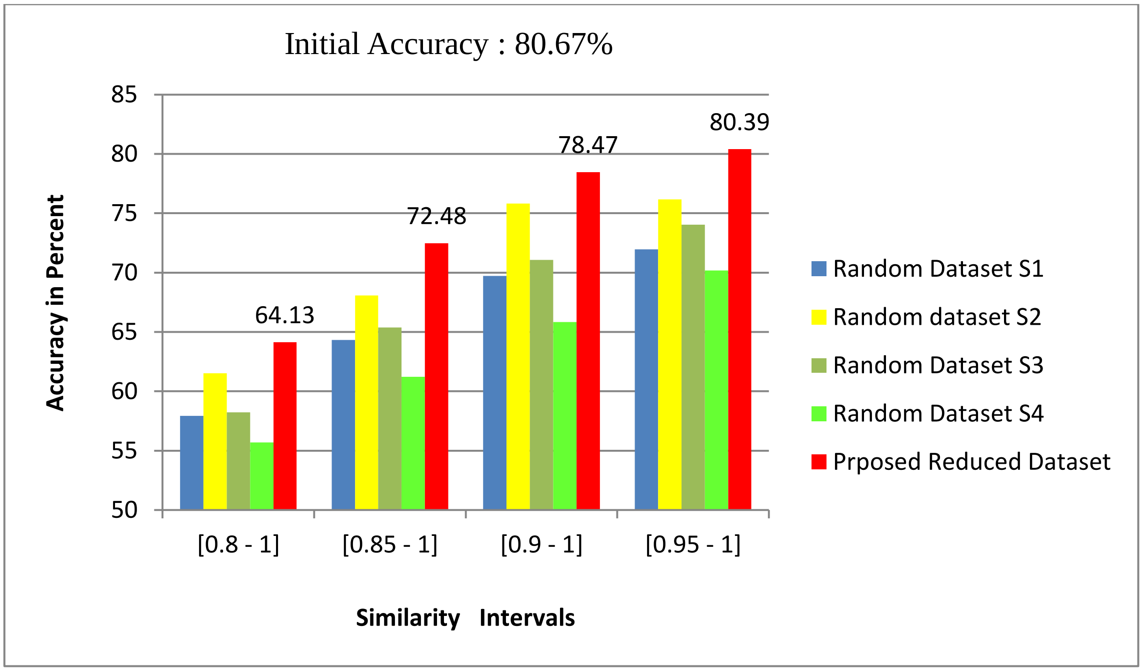


Figure 3 : Recognition results with training a FOCR system (letters) using the proposed reduced subsets and also using random subsets $\mathbf{S_1}$ to $\mathbf{S_4}$

Table 4 : Recognition time using different versions of reduced datasets (letters samples)

Similarity Interval for System Training	No. of Samples used for training the system using *k*-NN classifier	Recognition Time for a Sample (Seconds)	Ratio of Recognition Time to Initial Recognition Time T1
Using all training samples	70,645	T1 = 0.4456	T1/T1 = 100%
[0.95 – 1]	64,755	T2 = 0.4117	T2/T1 = 92.39%
[0.90 – 1]	56,613	T3 = 0.3640	T3/T1 = 81.69%
[0.85 – 1]	45,978	T4 = 0.2991	T4/T1 = 67.12%
[0.80 – 1]	34,293	T5 = 0.2261	T5/T1 = 50.74%

As it is seen in Tables 1 and 3, the proposed FOCR system achieved higher accuracy, when it used the training reduced dataset ***R_Dataset***, in compared to the system when it used training datasets $\mathbf{S_1}$ to $\mathbf{S_4}$, because: 1) The proposed algorithm selects a subset of the initial dataset with minimum similarity (maximum diversity); and 2) A *k*-NN classifier with more diverse training samples can differentiate the input samples better than that a *k*-NN with similar training samples (section 2.4.1). In other words, the results show that the samples in the reduced dataset using the proposed method have more diverse patterns than other subsets, $\mathbf{S_1}$ to $\mathbf{S_4}$.

Appendix VII

Most-used Features in OCR Applications

Table 1 : The initial feature set, the first reduced feature set, and the final reduced feature set, proposed by dimensionality reduction method 2S_SA, in this study

			Datasets					
			Hoda Digits Part		Hoda Characters Part		MNIST	
	Feature	Initial_S	F/D-S1	F/D-S2	F/C-S1	F/C-S2	E-S1	E-S2
1	Aspect Ratio (Safabakhsh and Adibi, 2005)	√	√	√	√	√	√	√
2	Image Area (Bui, 2004; Karic & Martinovic, 2013)	√	√	√	√	√	√	√
3	Image Perimeter (Impedovo, 2013)	√	√	√	√	√	√	√
4	Image Diameter (Impedovo, 2013)	√	√	√	√	√	√	√
5	Image Extent (Impedovo, 2013)	√	√		√			
6	Image Eccentricity (Impedovo, 2013)	√	√		√			
7	Image Solidity (Impedovo, 2013)	√	√	√	√	√	√	√
8	Euler Number (Impedovo, 2013)	√						
9	X Coordinate Centre of Mass (Harifi and Aghagolzadeh, 2005)	√	√	√	√			
10	Y Coordinate Centre of Mass (Harifi and Aghagolzadeh, 2005)	√	√		√			
11	Pixel distribution density in up halve of the normalized image (Enayatifar & Alirezanejad, 2011)	√	√	√	√	√	√	√
12	Pixel distribution density in down halve of the normalized image (Enayatifar & Alirezanejad, 2011)	√	√	√	√	√	√	√
13	Ratio of up halve pixels density to down halve pixel density (Shayegan et al., 2014)	√	√	√	√	√	√	√
14	Pixel distribution density in left halve of the normalized image (Dehghani et al. 2001)	√	√	√	√	√	√	√
15	Pixel distribution density in right halve of the normalized image (Dehghani et al. 2001)	√	√		√		√	√
16	Ratio of right halve pixels density to left halve pixel density (Pirsiyavash et al., 2005)	√	√		√		√	√
17	Pixel distribution density above the main diagonal of the normalized image (Soltanzadeh and Rahmati, 2004)	√	√	√	√	√	√	√
18	Pixel distribution density under the main diagonal of the normalized image (Soltanzadeh and Rahmati, 2004)	√	√	√	√	√	√	√
19	Ratio of image bounding box to number of foreground pixels (Mowlaei et al., 2002)	√	√	√	√	√	√	√
20	Variance of image vertical histogram (Soltanzadeh and Rahmati, 2004)	√	√	√	√	√	√	√

21	Variance of image horizontal histogram (Soltanzadeh and Rahmati, 2004)	√	√		√	√		
22	Normalized horizontal transition (Alaei et al., 2009b)	√	√	√	√	√	√	√
23	Normalized vertical transition (Alaei et al., 2009b)	√	√	√	√	√	√	√
24	Maximum horizontal crossing counts (Soltanzadeh and Rahmati, 2004)	√	√	√	√	√	√	√
25	Maximum vertical crossing counts (Soltanzadeh and Rahmati, 2004)	√	√		√	√		
26	Number of start points in image skeleton (Mozaffari et al., 2005b)	√	√	√	√	√	√	√
27	Number of end points in image skeleton (Mozaffari et al., 2005b)	√	√	√	√	√	√	√
28	Number of branch points in image skeleton (Mozaffari et al., 2005b)	√	√	√	√	√	√	√
29	Number of corner points in image skeleton (Mozaffari et al., 2005b)	√	√	√	√	√	√	√
30	Number of crossing points in image skeleton (Mozaffari et al., 2005b)	√	√	√	√	√	√	√
31	Location of start points in image skeleton (Mozaffari et al., 2005b)	√	√		√	√	√	√
32	Location of end points in image skeleton (Mozaffari et al., 2005b)	√	√		√	√	√	√
33	Location of crossing points in image skeleton (Rajabi et al., 2012)	√	√		√	√	√	√
34	Location of corner points in image skeleton (Rajabi et al., 2012)	√	√	√	√	√	√	√
35	Number of end points in zone 1 of 6-zoned image area (Rajabi et al., 2012)	√	√	√	√	√	√	√
36	Number of end points in zone 2 of 6-zoned image area (Rajabi et al., 2012)	√	√	√	√	√	√	√
37	Number of end points in zone 3 of 6-zoned image area (Rajabi et al., 2012)	√	√	√	√	√	√	√
38	Number of end points in zone 4 of 6-zoned image area (Rajabi et al., 2012)	√	√	√	√	√	√	√
39	Number of end points in zone 5 of 6-zoned image area (Rajabi et al., 2012)	√	√	√	√	√	√	√
40	Number of end points in zone 6 of 6-zoned image area (Rajabi et al., 2012)	√	√	√	√	√	√	√
41	Number of connected components (Shanbehzadeh et al., 2007)	√			√	√		
42	Number of foreground pixels in slices 1 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
43	Number of foreground pixels in slices 2 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
44	Number of foreground pixels in slices 3 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
45	Number of foreground pixels in slices 4 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
46	Number of foreground pixels in slices 5 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
47	Number of foreground pixels in slices 6 out of 12 of image in polar coordinates	√	√		√	√	√	

	(Pourasad et al., 2011)							
48	Number of foreground pixels in slices 7 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
49	Number of foreground pixels in slices 8 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
50	Number of foreground pixels in slices 9 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
51	Number of foreground pixels in slices 10 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
52	Number of foreground pixels in slices 11 out of 12 of image in polar coordinates	√	√		√	√	√	
53	Number of foreground pixels in slices 12 out of 12 of image in polar coordinates (Pourasad et al., 2011)	√	√		√	√	√	
54	Average of absolute value distance y from Bounding box's Y coordinate (Mowlaei et al., 2002)	√			√			
55	Average of multiplication distance x and square distance y from COM coordinates (Rajabi et al., 2012)	√			√			
56	Number of local maxima points in horizontal projections (Soltanzadeh and Rahmati, 2004)	√			√	√	√	
57	Number of local maxima points in vertical projections (Soltanzadeh and Rahmati, 2004)	√			√	√	√	
58	Normalized invariant central moments order 1 (Alirezaee et al., 2004b)	√	√		√	√	√	√
59	Normalized invariant central moments order 2 (Alirezaee et al., 2004b)	√	√		√	√	√	√
60	Normalized invariant central moments order 3 (Alirezaee et al., 2004b)	√	√		√	√	√	√
61	Normalized invariant central moments order 4 (Alirezaee et al., 2004b)	√	√		√	√		
62	Normalized invariant central moments order 5 (Alirezaee et al., 2004b)	√	√		√	√		
63	Normalized invariant central moments order 6 (Alirezaee et al., 2004b)	√	√		√	√		
64	Normalized invariant central moments order 7 (Alirezaee et al., 2004b)	√	√		√	√		
65	Top concavities in the image's skeleton (Khedher et al., 2005)	√	√	√	√	√	√	√
66	Down concavities in the image's skeleton (Khedher et al., 2005)	√	√	√	√	√	√	√
67	Right concavities in the image's skeleton (Khedher et al., 2005)	√	√	√	√	√	√	√
68	Left concavities in the image's skeleton (Khedher et al., 2005)	√	√	√	√	√	√	√
69	Number of modified horizontal transitions (Alaei et al., 2009b)	√	√	√	√	√	√	√
70	Number of modified vertical transitions (Alaei et al., 2009b)	√	√	√	√	√	√	√
71	Average of multiplication distance X from COM (Shanbehzadeh et al., 2007)	√	√		√	√	√	√
72	Average of multiplication distance Y from COM	√	√		√		√	√

	(Shanbehzadeh et al., 2007)							
73	Average of square distance x from x_bounding_box (Mowlaei et al., 2002)	√	√		√	√	√	√
74	Average of square distance y from y_bounding_box (Mowlaei et al., 2002)	√	√		√	√	√	√
75	Ratio of upper half variance to lower half variance of an image (Mozaffari et al., 2005b)	√			√		√	
76	Ratio of foreground pixels above the main diagonal to below the main diagonal (Soltanzadeh and Rahmati, 2004)	√	√	√	√	√	√	√
77	Ratio of pixel distribution of top-left image quarter to down-left image quarter (Shayegan et al., 2014)	√	√		√	√	√	
78	Ratio of number of foreground pixel in up-left quarter to image density (Shayegan et al., 2014)	√			√	√	√	
79	Ratio of number of foreground pixel in up-right quarter to image density (Shayegan et al., 2014)	√			√	√	√	
80	Ratio of number of foreground pixel in down-left quarter to image density (Shayegan et al., 2014)	√			√	√	√	
81	Ratio of number of foreground pixel in down-right quarter to image density (Shayegan et al., 2014)	√			√	√	√	
82	Ratio of number of foreground pixel above image main diagonal to under image main diagonal (Soltanzadeh and Rahmati, 2004)	√	√		√	√	√	√
83	Ratio of number of foreground pixel above image off diagonal to under image of diagonal (Soltanzadeh and Rahmati, 2004)	√	√	√	√	√	√	√
84	Ratio of mean of y coordinates foreground pixels to mean of x coordinates foreground pixels (Sahyegan et al., 2014)	√	√	√	√	√	√	√
85	Mean of distance between x coordinate of foreground pixels of x coordinate of COM (Alirezaee et al., 2004a)	√			√	√		
86	Mean of distance between y coordinate of foreground pixels of y coordinate of COM (Alirezaee et al., 2004a)	√						
87	Number of secondary components (Shanbehzadeh et al., 2007)	√			√	√		
88	Relative location of secondary components to main body (Shanbehzadeh et al., 2007)	√			√	√		
89	Simple loops (Safabakhsh and Adibi 2005)	√	√	√	√	√	√	√
90	Complex loops (Safabakhsh and Adibi 2005)	√			√	√	√	√
91	DCT coefficient (1,1) related to the main image (Mowlaei et al., 2002)	√	√	√	√	√	√	√
92	DCT coefficient (1,2) related to the main image (Mowlaei et al., 2002)	√	√	√	√	√	√	√
93	DCT coefficient (1,3) related to the main image (Mowlaei et al., 2002)	√					√	√
94	DCT coefficient (1,4) related to the main image (Mowlaei et al., 2002)	√	√	√	√	√	√	√
95	DCT coefficient (1,5) related to the main image (Mowlaei et al., 2002)	√	√	√	√	√	√	√
96	DCT coefficient (2,1) related to the main image (Mowlaei et al., 2002)	√	√	√	√		√	√
97	DCT coefficient (2,2) related to the main image (Mowlaei et al., 2002)	√			√			
98	DCT coefficient (2,3) related to the main image (Mowlaei et al., 2002)	√			√			
99	DCT coefficient (2,4) related to the main image (Mowlaei et al., 2002)	√	√	√	√		√	√
100	DCT coefficient (2,5) related to the main image (Mowlaei et al., 2002)	√			√	√		

101	DCT coefficient (3,1) related to the main image (Mowlaei et al., 2002)	√			√	√	√	
102	DCT coefficient (3,2) related to the main image (Mowlaei et al., 2002)	√			√	√	√	
103	DCT coefficient (3,3) related to the main image (Mowlaei et al., 2002)	√						
104	DCT coefficient (3,4) related to the main image (Mowlaei et al., 2002)	√						
105	DCT coefficient (3,5) related to the main image (Mowlaei et al., 2002)	√						
106	DCT coefficients (1,1) for image’s outer boundary (Mowlaei and Faez, 2003)	√	√	√	√	√	√	√
107	DCT coefficients (1,2) for image’s outer boundary (Mowlaei and Faez, 2003)	√	√	√	√	√	√	√
108	DCT coefficients (1,3) for image’s outer boundary (Mowlaei and Faez, 2003)	√			√		√	√
109	DCT coefficients (1,4) for image’s outer boundary (Mowlaei and Faez, 2003)	√			√		√	√
110	DCT coefficients (1,5) for image’s outer boundary (Mowlaei and Faez, 2003)	√			√			
111	DCT coefficients (2,1) for image’s outer boundary (Mowlaei and Faez, 2003)	√	√	√	√		√	√
112	DCT coefficients (2,2) for image’s outer boundary (Mowlaei and Faez, 2003)	√			√		√	√
113	DCT coefficients (2,3) for image’s outer boundary (Mowlaei and Faez, 2003)	√			√		√	√
114	DCT coefficients (2,4) for image’s outer boundary (Mowlaei and Faez, 2003)	√					√	√
115	DCT coefficients (2,5) for image’s outer boundary (Mowlaei and Faez, 2003)	√						
116	DCT coefficients (3,1) for image’s outer boundary (Mowlaei and Faez, 2003)	√					√	√
117	DCT coefficients (3,2) for image’s outer boundary (Mowlaei and Faez, 2003)	√					√	√
118	DCT coefficients (3,3) for image’s outer boundary (Mowlaei and Faez, 2003)	√	√	√			√	√
119	DCT coefficients (3,4) for image’s outer boundary (Mowlaei and Faez, 2003)	√	√	√			√	√
120	DCT coefficients (3,5) for image’s outer boundary (Mowlaei and Faez, 2003)	√						
121	DCT coefficients for image profiles up to order 5 (Mowlaei and Faez, 2003)	√						
122	DCT coefficients (1,1) of image up-profile (Alessandro and Koerich, 2003)	√	√	√	√	√	√	√
123	DCT coefficients (1,2) of image up-profile (Alessandro and Koerich, 2003)	√	√	√	√	√	√	√
124	DCT coefficients (1,3) of image up-profile (Alessandro and Koerich, 2003)	√			√			
125	DCT coefficients (1,1) of image down-profile (Alessandro and Koerich, 2003)	√	√	√	√	√	√	√
126	DCT coefficients (1,2) of image down-profile (Alessandro and Koerich, 2003)	√	√	√	√	√	√	√
127	DCT coefficients (1,3) of image down-profile (Alessandro and Koerich, 2003)	√			√	√	√	
128	DCT coefficients (1,1) of image left-profile (Alessandro and Koerich, 2003)	√	√	√	√		√	√
129	DCT coefficients (1,2) of image left-profile	√	√	√	√		√	√

	(Alessandro and Koerich, 2003)							
130	DCT coefficients (1,3) of image left-profile (Alessandro and Koerich, 2003)	√			√		√	
131	DCT coefficients (1,1) of image right-profile (Alessandro and Koerich, 2003)	√	√	√	√		√	√
132	DCT coefficients (1,2) of image right-profile (Alessandro and Koerich, 2003)	√	√	√	√		√	√
133	DCT coefficients (1,3) of image right-profile (Alessandro and Koerich, 2003)	√			√			