A NEW REPRESENTATION FOR MATCHING WORDS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Esra Ataer

July, 2007

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Pınar Duygulu Şahin(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Selim Aksoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fatoş Yarman-Vural

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

A NEW REPRESENTATION FOR MATCHING WORDS

Esra Ataer
M.S. in Computer Engineering
Supervisor: Assist. Prof. Pınar Duygulu Şahin
July, 2007

Large archives of historical documents are challenging to many researchers all over the world. However, these archives remain inaccessible since manual indexing and transcription of such a huge volume is difficult. In addition, electronic imaging tools and image processing techniques gain importance with the rapid increase in digitalization of materials in libraries and archives. In this thesis, a language independent method is proposed for representation of word images, which leads to retrieval and indexing of documents. While character recognition methods suffer from preprocessing and overtraining, we make use of another method, which is based on extracting words from documents and representing each word image with the features of invariant regions. The bag-of-words approach, which is shown to be successful to classify objects and scenes, is adapted for matching words. Since the curvature or connection points, or the dots are important visual features to distinct two words from each other, we make use of the salient points which are shown to be successful in representing such distinctive areas and heavily used for matching. Difference of Gaussian (DoG) detector, which is able to find scale invariant regions, and Harris Affine detector, which detects affine invariant regions, are used for detection of such areas and detected keypoints are described with Scale Invariant Feature Transform (SIFT) features. Then, each word image is represented by a set of visual terms which are obtained by vector quantization of SIFT descriptors and similar words are matched based on the similarity of these representations by using different distance measures. These representations are used both for document retrieval and word spotting.

The experiments are carried out on Arabic, Latin and Ottoman datasets, which included different writing styles and different writers. The results show that the proposed method is successful on retrieval and indexing of documents even if with different scripts and different writers and since it is language independent, it can be easily adapted to other languages as well. Retrieval performance of the system is comparable to the state of the art methods in this field. In addition, the system is succesfull on capturing semantic similarities, which is useful for indexing, and it does not include any supervising step.

Keywords: word matching, document retrieval, bag-of-features.

ÖZET

KELİME EŞLEME YÖNTEMİ İÇİN YENİ BİR NİTELEME

Esra Ataer
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Assist. Prof. Pınar Duygulu Şahin
Temmuz, 2007

Tarihi arşivler dünyanın pek cok yerinden araştırmacının ilgi alanına girmektedir. Fakat, belgelerin elle çevirisi ve dizinlemesi zor bir iş olduğu için bu arşivler kullanılamaz durumdadır. Ayrıca elektronik imgeleme araçları ve imge işleme teknikleri kütüphane ve arşivlerin dijital ortama aktarılmasıyla gün geçtikçe önem kazanmaktadır. Bu tezde erişim ve dizinlemede kullanılmak üzere kelime imgelerini nitelemek için dilden bağımsız bir çözüm getirilmektedir. Karakter tanıma teknikleri aşırı önişleme ve öğrenme yönünden eksiklikler içerirken, önerilen yöntem belgeleri kelimelere bölütleyerek ayırt edici bölgeleri kullanarak bu kelimeleri nitelemektedir. Nesne ve manzara tasnifinde başarı gösteren görsel-öğeler-kümesi yöntemi kelime eşlemeye uyarlandı. Kıvrım, bağlantı bölgeleri ve noktalar kelimeyi ayırt etmek için öenmli görsel öznitelikler olduğu için bu bölgeleri tanımlamada başarılı olan ve imge eşlemede sıkça kullanılan taç noktalar kullanıldı. Bu bölgelerin tespit edilmesinde Gauss Farkı ve Harris-Affine sezicilerinden yararlanıldı ve tespit edilen bölgeler Scale Invariant Feature Transform (SIFT) öznitelikleriyle tanımlandı. Her kelime SIFT tanımlayıcılarının vektör nicemlenmesiyle oluşturulan görsel öğelerin değişik dağılımlarına göre nitelendi ve bu niteleme belge erişim ve dizinlemesi için kullanıldı.

Deneyler farklı yazı tipi içeren ve çeşitli yazarlarca yazılmış Arapça, Latince ve Osmanlıca belgelerde gerçekleştirildi. Veri kümelerinin farklı yazı tipleri içermesine ve çeşitli yazarlarca oluşturulmuş olmasına rağmen, sonuçlar önerilen sistemin belge erişimi ve dizinlemede başarılı olduğunu göstermektedir. Önerilen yöntem dilden bağımsız olduğu için kolayca başka dillere de uyarlanabilir. Sistem belge erişiminde bu alandaki en iyi yöntemlere yakın bir başarım sergilemektedir. Bunun yanında önerilen yöntemin anlamsal benzerlikleri bulmada başrılı olması belge dizinleme için etkili biçimde kulanılabileceğini göstermektedir.

Anahtar sözcükler: kelime eşleme, belge erişimi, görsel öğeler kümesi.

Acknowledgement

I would like to express my gratitude to my supervisor Assist. Prof. Pınar Duygulu Şahin who has guided me throughout my studies.

I would like to express my special thanks to Assist. Prof. Selim Aksoy and Prof. Dr. Fatoş Yarman-Vural for showing keen interest to the subject matter and accepting to read and review the thesis.

I am thankful to RETINA Learning and Vision Group for their comments and suggestions on my studies.

Finally my greatest gratitude is to my family (my father and mother, my morale source Atahan, sweet Merve and little Mustafa), who are the reason of the stage where I am standing now.

Contents

List of Figures

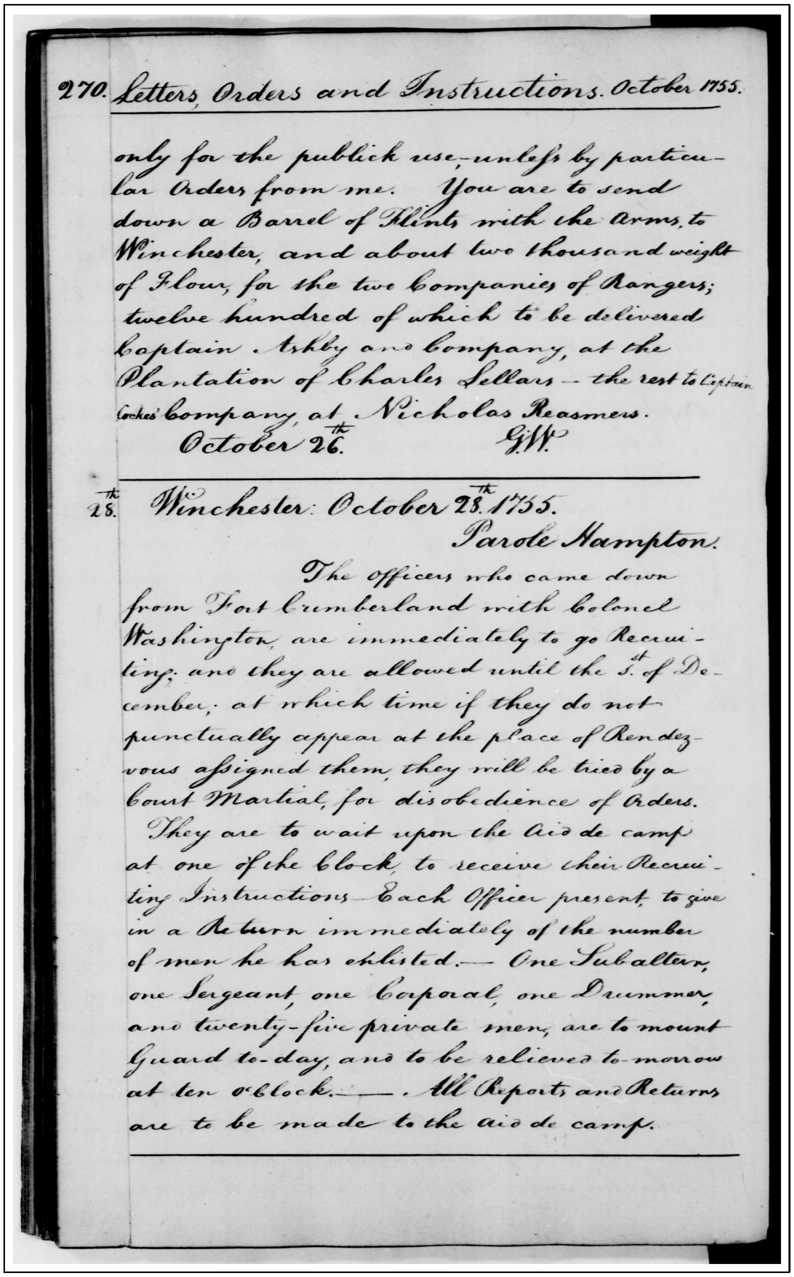
List of Tables

Chapter 1

Introduction

Large archives of historical documents are in the scope of many researchers from all over the world. Retrieval and recognition of documents in these historical and cultural archives are of crucial importance due to increasing demand to access to the valuable content stored. Manual indexing and transcription of these documents are rather labor intensive, supporting the strong need to perform these jobs without human intervention.

Although document recognition is a long standing problem, current document processing techniques, which are mostly based on character recognition [34, 19, 60, 56, 11, 4, 5], does not present accurate results for historical documents, in deteriorating conditions with poor quality (see Figure 1.1). Also document recognition is still a challenge especially for connected scripting languages like Arabic and Ottoman [3, 13, 5, 30, 10, 4, 5, 44, 50, 53, 61, 7, 8, 40]. The alphabets of these languages are different from Latin alphabet and the problem gets more difficult, if the documents are handwritten and has various writers and writing styles.



270. Letters, Orders and Instructions. October 1755.

only for the publick use,–unleſs by particular Orders from me. You are to send down a Barrel of Flints with the Arms, to Winchester, and about two thousand weight of Flour, for the two Companies of Rangers; twelve hundred of which to be delivered Captain Ashby and Company, at the Plantation of Charles Sellars – the rest to Captain Cockes' Company, at Nicholas Reasmers.

October 26th. G.W.

28th. Winchester: October 28th 1755.

Parole Hampton.

The officers who came down from Fort Cumberland with Colonel Washington, are immediately to go Recruiting; and they are allowed until the 1st of December; at which time if they do not punctually appear at the place of Rendezvous aſsigned them, they will be tried by a Court Martial, for disobedience of Orders.

They are to wait upon the Aid de camp at one of the Clock, to receive their Recruiting Instructions–Each Officer present, to give in a Return immediately of the number of men he has enlisted.— One Subaltern, one Sergeant, one Corporal, one Drummer, and twenty-five private men, are to mount Guard to-day, and to be relieved to-morrow at ten o'Clock.——. All Reports and Returns are to be made to the aid de camp.

Figure 1.1: An example page from George Washington collection of Library of Congress. This collection includes historical letters written by George Washington. Note that the document is in poor condition that makes it harder to recognize.

Recognition of Ottoman characters is extremely challenging, if not impossible, with the traditional character recognition systems, due to the elaborate skewed and elongated characteristics of the calligraphy. Some firman (royal document) examples can be seen on Figure 1.2. As can be seen from the figure the documents resemble pictures rather than ordinary texts and segmentation of characters from documents is truly a formidable problem.

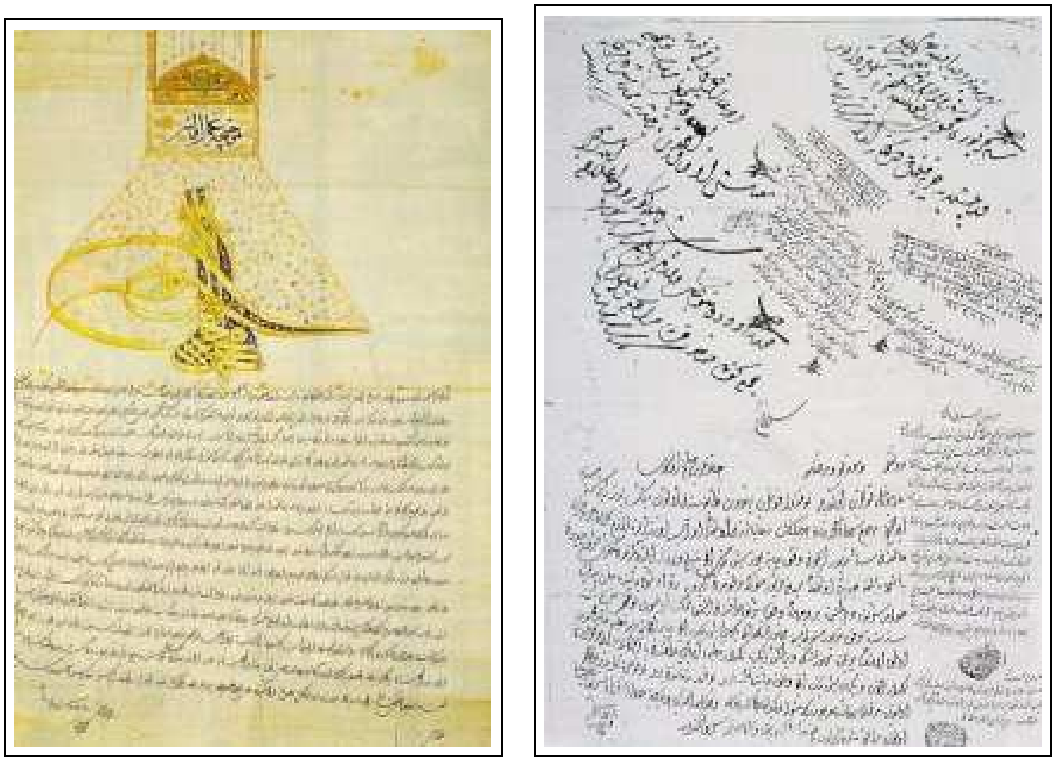


Figure 1.2: Two firman examples (royal documents) from Ottoman Emperor [43]. On the right, there exists a Tuğra (signature of the Sultan) at the top of the document. As illumination (*tezhip*) is also a respected art in Ottoman era, tuğra is illuminated like many other documents. On the right, some parts of the document is written from bottom right to upper left, but some parts are written from right to left directly. Note that the documents are very confusing and resemble to paintings rather than writing.

Consequently, the observations show that the problem can not be solved with current character recognition techniques effectively. Recently, Manmatha *et al.* approached the problem differently and used the idea of word matching on historical manuscripts as they make use of some global features of word images. Their system uses pixel based features, which are generally used in traditional character recognition systems.

We further follow the word matching methodology, but instead of classical features used on document recognition, we make use of features, that are mainly used for image matching. Similar to an illiterate novice person, who is unacquainted to a particular language and unaware of the shape of letters, we assume that each word is analogous to an image rather than a collection of characters and this way, we approach the problem as an image matching problem instead of a character recognition problem. In addition curvature, connection points or dots of a word image are important visual feature in order to distinguish it from others. So we make use of the salient points which are shown to be successful in representing such distinctive areas and heavily used for matching. The proposed representation is language independent and successfull on finding similarity between two words without a heavy modeling or supervising step. Retrieval performance of the system is comparable to the state of the art techniques in this field and performs even better in simplicity and running time.

1.1 Overview of the proposed method

With the major assumption, a word is an image rather than a collection of characters, we need a good representation for each word-image, so that we can easily make it distinct from the others. As we approach the problem analogous to a novice person, who does not speak a particular language and does not recognize the shapes of the individual letters, visual feature points like connection or curvature points or dots of a word have importance for distinguishing it from the other word images. So, in this study, we make use of interest points for representation of word images.

The information coming from interest points give strong cues for recognition of a word image. However, we still don't have an exact representation for a word image and we have to effectively use interest point information so that we have a distinctive representation for each word image. The solution idea comes from the text retrieval systems, which sees each documents as a collection of words and identifies each document with its existing terms. Similarly, we can treat a word image as a collection of visual words, which are generated by vector quantization of keypoint descriptors. By this way, we create a visual codebook for the word image dataset and identify each word with its existing visual terms from this codebook. Then, we propose various representations for word images either with different distributions of these visual words or string representation of the visual vocabulary.

Since we have a distinctive representation for each word, we used it for retrieval and indexing purposes throughout the experiments. By this way, we describe a language independent solution for representing word images, so that it gives rise to effective and efficient retrieval and indexing of handwritten documents.

With the focus on the problem of retrieval and indexing of documents of connected scripting languages, the experiments are carried out on Ottoman manuscripts of printed and handwritten type. However, since accessing Ottoman archives is difficult for us as to many researchers and there is not any convenient Ottoman dataset with ground truth information, we prefer to run some detailed experiments on an Arabic dataset, which have numerous writers and writing styles with a utilizable ground truth information. The observation that Ottoman and Arabic script share many common properties, play role on choosing a dataset of Arabic script for performing the additional experiments. George Washington collection of Library of Congress, including many historical correspondences written by a single author, offer a good example to apply our method, therefore we also perform retrieval and indexing for this dataset and see that results are comparable with the most recent word matching techniques. In addition, this dataset spectrum provide the opportunity to show the language independence of the proposed method.

1.2 Organization of the Thesis

Chapter 2 of the thesis, discusses related studies about the subject, where the approaches are explained in a comparative manner with the proposed method.

Chapter 3 of the thesis introduces the main contribution of the thesis, which is representation of word images. Representation is explained in four steps that are detection and description of visual points, creation of visual terms and representation of word images.

Chapter 4 overviews the retrieval and indexing steps that make use of the proposed representation. Distance measures and classification types used in retrieval and word spotting are listed and discussed in this chapter.

Experimental evaluation of the proposed method is presented in Chapter 6 in which its comparison with some other techniques is also given and interpreted.

Chapter 7 reviews the results and the contributions of this thesis and outlines future research directions on this subject.

Chapter 2

Related Work

Offline handwriting recognition has enabled many applications like searching on large volumes of documents, postal mail sorting or transcription of scanned text documents. While these applications require good quality images, the quality of historical documents is significantly degraded. Hence, historical document processing is a rather difficult problem, which should be approached in a different manner than traditional offline handwriting approaches. In this thesis, we propose a general solution for the problem, independent from the scripting type. Proposed method makes use of word matching idea by representing each word image with its salient regions.

This chapter presents a general overview on offline handwriting recognition and specifically word matching techniques. It also underlines some techniques on object recognition and scene classification, which we adapted to our problem, in our case word recognition.

2.1 Document Retrieval and Recognition

As explained in [30], general components of a document recognition technique are preprocessing, representation, character/subword/word segmentation, feature detection and recognition. These steps may not be consecutive and some approaches do not use all of the steps but only a subset of them.

Preprocessing step aims to clean and eliminate noise and make the document ready for feature extraction. A document is represented according to its extracted features and recognizer runs based on the representation of the document. Current techniques on document recognition are categorized into three groups according to the part that they make recognition on; character based, subword/stroke based and word based techniques. These techniques extracts the related parts (characters, subwords/strokes or words) from documents and tries to make recognition on them. Thus recognizers are generated according to the part which they will process on.

There are many recognition techniques for handwriting like Artificial Neural Networks, Hidden Markov models etc. The recognizers are mainly modeled according to the segmented unit and representation of the documents. Hence representing the document is an important task for a document recognition system.

Character based system are based on the representation and recognition of individual characters. Thus they firstly aim to recognize the characters resulting in the recognition of whole documents. But since character shapes vary for handwriting, this kind of methods require a huge amount of data in order to generate and train a robust model and suffer from not covering the feature space effectively.

After this general overview on document recognition, recent studies on this field will be overviewed in the following.

In [1] Abuhaiba *et al.* generates an Arabic character recognition system, in which each character is represented with the tree structure of its skeleton.

Gillies *et al.*[16] make use of atomic segments and Viterbi algorithm in order to create an Arabic text recognition system.

Edwards *et al.* [14] described a generalized HMM model in order to make a scanned Latin manuscript accessible to full text search. The model is fitted by using transcribed Latin as a transition model and each of 22 Latin letters as the emission model. Since Latin letters are isolated and Latin is a regular script, their system is capable of making a successful full text search, but the proposed system is not be fitted to another scripting style directly since they make training with Latin letter models.

Chan *et al.* [13] presented a segmentation based approach that utilizes gHMMs with a bi-gram letter transition model. Their lexicon-free system performs text queries on off-line printed and handwritten Arabic documents. But even if their system is successful on searching Arabic printed and handwriting documents, it is language dependent since it is segmentation based. In addition the documents have gone to a heavy preprocessing step, where character overlaps and diacritics are removed.

Saykol *et al.* [50] used the idea of compression for content-based retrieval of Ottoman documents. They create a code book for the characters and symbols in the data set and processed the queries based on compression according to this codebook. Scale invariant features named distance and angular span are used in the formation of the codebook.

Schomaker *et al.* [51] used some geometrical features for describing cursive handwritten characters.

Mozaffari *et al.* [38] developed a structural method embedded with statistical features for recognition of Farsi/Arabic numerals, where they used standard feature points to decompose the character skeleton into primitives. Some statistical measures are used to statistically describe the direction and the curvature of the primitives.

Belongie *et al.* [11] proposed a framework for measuring the similarity between two shapes, which can be also used for digit recognition. Their approach is based on using correspondences to estimate an alignment transform between two shapes. Arica *et al.* [6] introduced a different shape descriptor, that makes use of Beam Angle Statistics. Their scale, rotation and translation invariant shape descriptor is defined by the third order statistics of the beam angles in the area.

One of the most popular methods in historical document recognition is word matching idea introduced by Manmatha *et al* [32]. The next section overviews some studies on this approach.

2.2 Techniques on Word Matching

Recently, Rath and Manmatha [48] proposed a word-image matching technique for retrieval of historical documents by making use of Dynamic Time Warping (DTW) and show that the documents can be accessed effectively without requiring recognition of characters with their word spotting idea. They use intensity, background-ink transition, lower and upper bound of the word as the features for matching process. They make experiments with seven different methods, which showed that their projection based DTW method results in the highest precision [33]. In their another study [49] they used a Hidden Markov Model based automatic alignment algorithm in order to align text to handwritten data. Experiments are done on the same dataset as before and they claim that this algorithm outperforms the previous DTW approach.

Srihari *et al.* [55] proposed a system using word matching idea after a prototype selection step. They make use of 1024 binary features in word matching step and acquired promising results for a dataset with various writers.

In [52], matching on bywords is proposed by using different combinations of five feature categories: angular line features, co-centric circle features, projection profiles, Hu's moment and geometric features.

Benouareth *et al.* [12] presented an off-line segmentation free Arabic words recognition system by generating discrete Hidden Markov Models with explicit state duration of various kinds (Gauss, Poisson and Gamma) for the word classification purpose. The experiments are carried out on IFN-ENIT database [45] of handwritten Tunisian city names and show the comparative performance of HMMs with different types.

Adamek *et al.* [2] use the idea of word matching in order to index historical manuscripts. They make use of contour based features rather than profile based features for representing word images.

Konidaris *et al.* [23] proposed a retrieval system that is optimized by user feedback. Their system performs retrieval after segmenting documents into words and representing each word image with extracted features.

In our previous study [7], Ottoman words are matched using a sequence of elimination techniques which are mainly based on vertical projection profiles. The results show that even with simple features promising results can be achieved with word matching approach.

The proposed method makes use of word matching idea with an adaption of Bag-of-Features (BoF) approach. Hence the next section introduce BoF approach as an image matching technique and summarizes the studies made on this field.

2.3 Bag-of-Features Approach (BoF)

The problem of classifying objects and scenes according to their semantic content is currently one of the hardest challenges with the difficulties like pose and lighting changes and occlusion. While global features does not overcome these difficulties effectively, BoF approach which captures the invariance aspects of local keypoints has recently attracted various research attentions.The main idea of BoF is to represent an image as a collection of keypoints. In order to describe BoF, a visual vocabulary is constructed through a vector quantization process on extracted keypoints and each keypoint cluster is treated as a ”visual term” (visterm) in the visual vocabulary. Although it is a simple technique and does not use geometry information, it has demonstrated promising results on many visual classification tasks [54, 26, 39].

There are six popular detectors for detection of keypoints, Laplacian of Gaussian (LoG) [29], Difference of Gaussian (DoG) [31], Harris Laplace and Harris Affine [35], Hessian Laplace and Hessian Affine [37]. LoG detector detects blob-like regions by generating a scale space representation by successively smoothing the image with Gaussian based kernels of different sizes. Then local maxima's are detected as keypoints. DoG is studied by Lowe, who computed Difference of Gaussian (DoG) of images in different scales and found the points that show a local maxima/minima with its neighboring pixels, that are chosen as keypoints. DoG is an approximate and more efficient version of LoG. Harris and Hessian Laplace detector make use of Harris function and Hessian determinant respectively to localize points in scale space and selects the points that reach a local maxima over LoG. Harris Affine and Hessian Affine extended Harris and Hessian Laplace detector respectively to deal with significant affine transformations.

Mikolajczyk *et al.* presented a comparative study on keypoint descriptors [36]. They used many descriptors, that are Scale Invariant Feature Transform, gradient location and orientation histogram (GLOH), shape context, PCA-SIFT, spin images, steerable filters, differential invariants, complex filters, moment invariants, and cross-correlation of sampled pixel values. After the evaluation they conclude that independent of detector type, SIFT based descriptors performs best among all.

After detection and description of keypoints, the visual vocabulary is generated via clustering the documents into groups. Thus, like a document composing of words, an image can also be treated as a collection of visterms. Exact representation of images can differ according to the usage of visterm information. Visterm information can be used to create either the classical histogram of visterms or histograms with different weighting information.

Term weighting have critical impact on text retrieval system. Term Frequency Inverse Document Frequency (*tf-idf*) is one of the mostly used technique. As the name implies, it increase the importance of rarely appearing terms while decreasing the weight of usual appearing terms, in other words, stop words.

Current BoF approaches generally make use of tf-idf representation aiming to make rare visterms more distinctive and representative than the others [54, 26].

Jiang *et al.* presented a different representation style, namely Soft Weighting Scheme. In their method, each visterm weights the histogram according to its distance to the visterm centroid, where each keypoint weights not only the nearest visterm, but also top-N nearest visterms. They observed that their soft weighting scheme performs well independent of the vocabulary size.

Chapter 3

Representation of Word Images

With the main view that, a word is an image rather then a set of characters and interest points of a word image are important visual features to distinguish it from others, we propose a novel approach for representation of word images based on interest points, which will be later used for retrieval and recognition purposes.

This chapter introduces the main contribution of this thesis, which is generation of a language independent representation for word images. Firstly, interest points of the word image, that are important visual features, are extracted and described with the visual descriptors. Then extracted descriptors are fed into a vector quantization process which results in generation of visual terms. Each word is then represented by a collection of visual terms which is referred as bag of visual terms. Exact representation of a word image can either be done with different histogram distributions or string representation of its existing visual words.

In summary, the representation task is performed in four consecutive steps that are detection and description of keypoints, generation of visual terms and representation of word images as will be explained in related sections throughout the chapter.

3.1 Detection of Visual Points

An object in a database of images is represented both with local features or global features. Some local points in the image have greater importance with its invariance to scale and transformation. Hence, an image can be identified with the description of these keypoints. This kind of representation is mostly used in object recognition, scene classification and image matching [26, 36, 39, 54].

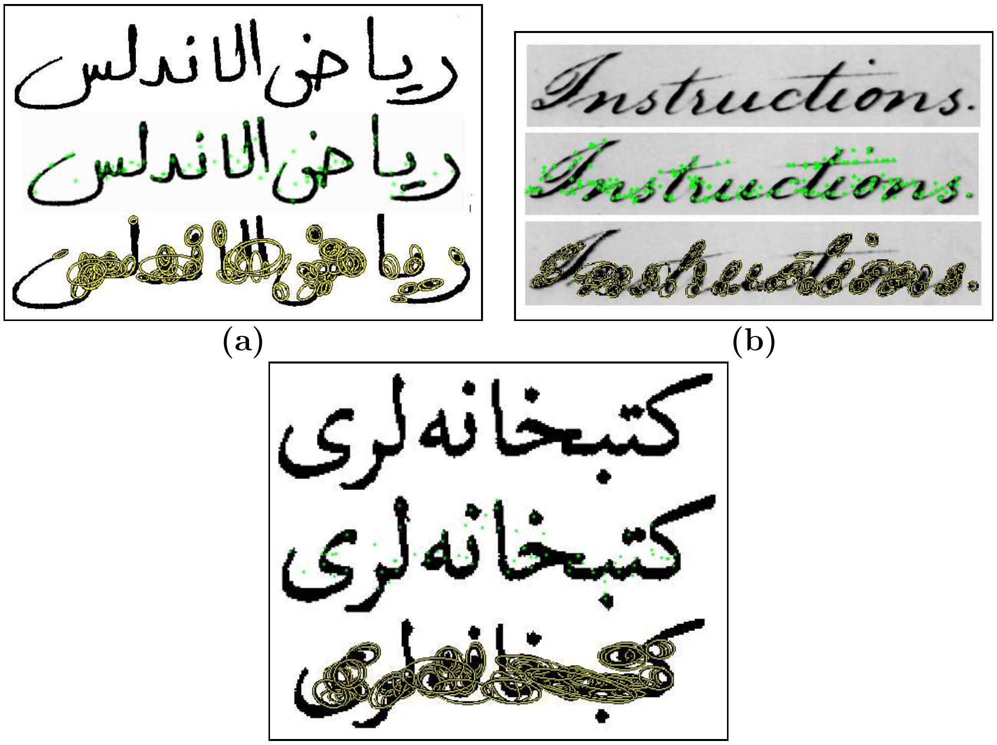
Detection of interest points is a challenging problem and in the scope of many studies [29, 31, 35, 37, 22]. Out of these detectors, we mainly stress on Lowe's Difference of Gaussian (DoG) detector and Mikolajczyk's Harris Affine detector, which are shown to be successful in numerous studies as they are briefly explained in the following.

Lowe followed four major stages in order to detect keypoints and generate sets of image features; scale-space extrema detection, keypoint localization, orientation assignment and finally keypoint descriptors [31]. In the first step, he computed difference of Gaussian (DoG) of images in different scales and found the points that show a local maxima/minima with its neighboring pixels, that are chosen as keypoint candidates. Secondly, keypoints are selected based on measures of their stability and localized with a detailed model determining location and scale. Then, orientation assignment of a keypoint is done according to its local gradient directions at the third stage. Finally, local image gradients are computed at the assigned scale in the region around each keypoint and used as the descriptor of the keypoint, which is referred as Scale Invariant Feature Transform (SIFT).

Mikolajczyk *et al.* extended Harris-Laplace detector to deal with significant affine transformations [35]. They used Harris multi-scale detector for detection of keypoints, where they assigned an initial location and scale for the keypoint simultaneously. To obtain the shape adaptation matrix for each interest point, they compute the second moment descriptor with automatically selected integration and derivation scale, where integration scale is the extrema over scale of normalized derivatives and derivation scale is the maximum of normalized isotropy.

Consequently, they acquire the points of an image that are invariant to affine transformations.

As we assumed that the curvature, connection points and the dots are important visual features of a word image, we observe that DoG detector and Harris Affine detector find such points of a word. Figure 3.1 shows detected keypoints of example word images of different scripts. Notice that detected points are usually curvature or connection points or dots independent from the type of script.



(a) **(b)**

(c)

Figure 3.1: Detected keypoints of three example words from **(a)** Arabic, **(b)** George Washington and **(c)** Ottoman datasets. For each part first image is the image itself, second are points detected with DoG detector and the third one shows affine invariant regions detected with Harris Affine detector.

In our study we make use of both detectors, but generally emphasized on Harris-Affine detector, since it gives better results as told in Chapter 5. The next section will explain the next step, that is description of the detected keypoints.

3.2 Description of Visual Points

In [36] different descriptor types are evaluated and the results show that independent of detector type, SIFT based descriptors perform best. Thus, we used SIFT descriptors in order to represent the detected keypoints.

SIFT descriptor is created by first computing the gradient magnitude and orientation at each point around the keypoint [31]. Then 8 bin orientation histogram of 4x4 subregion around the detected keypoint is computed resulting in a 128 element feature vector for that keypoint. Figure 3.2 illustrates the creation of SIFT descriptor for an interest point.

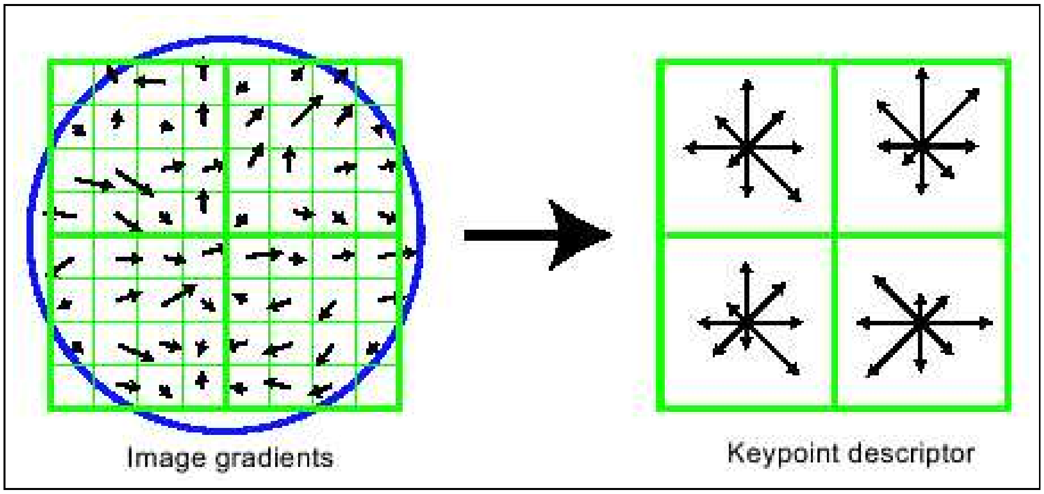


Figure 3.2: On the left are the gradient magnitude and orientation of the points in the region around the keypoint, which are weighted by a Gaussian window, indicated by the overlaid circle. On the right is the orientation histogram summarizing the contents over 4x4 subregions, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region[31].

For affine covariant regions, additional 5 more parameters, that defines the affine region, are appended to 128 element SIFT descriptor, resulting in a 133 element size feature vector. The first two parameters (u, v) are center of elliptic region and remaining three parameters (a, b, c) defines the affine transformation. The point (x, y) in the affine covariant region must satisfy the equation

$$a(x-u)(x-u)+2b(x-u)(y-v)+c(y-v)(y-v)=1 \tag{3.1}$$

where u, v, a, b, c are the additional parameters defining the region. Thus, we also make use of location information particularly, when we used Harris Affine detector.

After detection and description of keypoints of the word images, we proposed a compact representation for words by using bag-of-visterms approach. The next section will describe the visual term generation process, which is the main step of bag-of-visterms approach.

3.3 Visterm Generation

After representation of detected regions, the visual term, *visterm*, generation begins. This step is the most important step for the representation of word images since each word image is represented with visterm information.

Since number of detected keypoints are not much for a word image, there are not so much matching keypoints between the instances of a word. So, rather than matching words based on similarity of individual keywords, we prefer to use bag-of-words approach where the images are treated as documents with each image being represented by a histogram of visual terms. We also done experiments, based on the similarity of individual keypoints, but as reported in Chapter 5 it does not overcome the performance of the proposed method .

The visual terms, usually referred as *visterms*, are obtained by vector quantization of the feature vectors, the descriptors of detected keypoints. We run k-means on descriptors of keypoints extracted from each dataset separately and acquired respective visterms for the datasets. Finally, we have k visterm for each dataset and each keypoint is referred as one of these visterms.

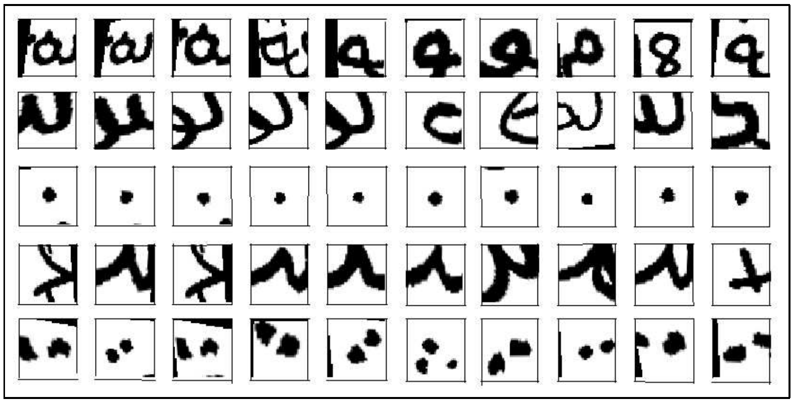


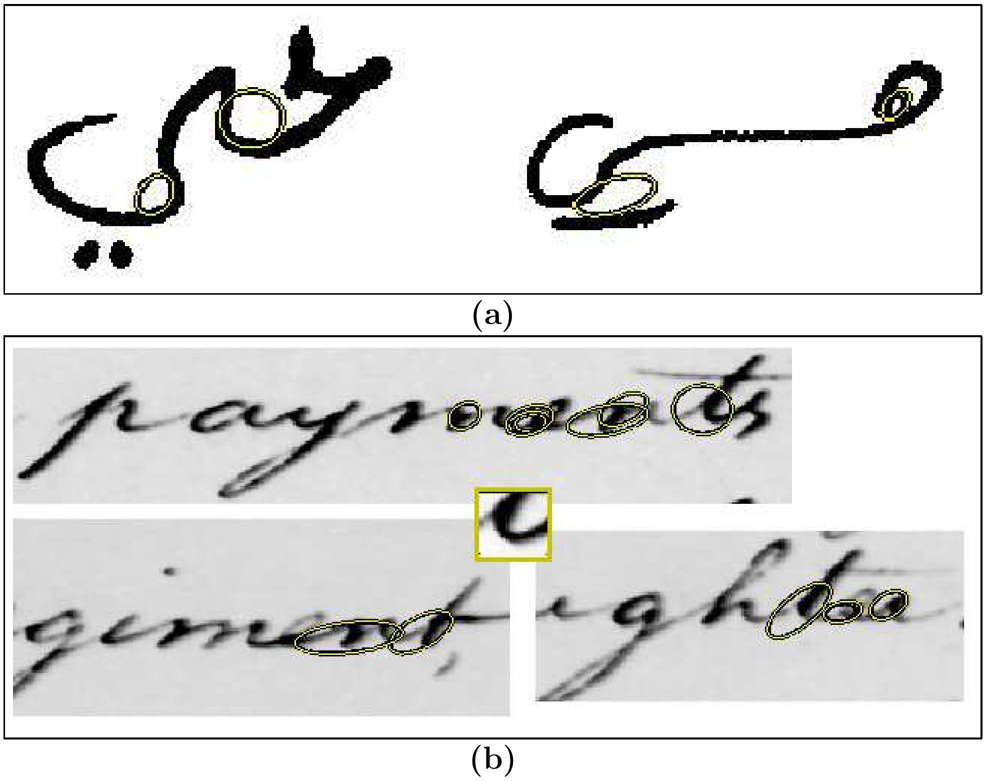
Figure 3.3: 10 example patches of five different visterms on Arabic dataset. Similar patches are grouped into same visterm cluster.

Patches of the same visterm look like each other and differs from the other visterms as seen in Figure 3.3. Figure 3.4 shows some visterms on different instances of a character on George Washington and a subword from Arabic Dataset. For different writings of the same character, detected keypoints belong to the same visterm, showing that a meaningful visual vocabulary is generated. An experiment for estimating the optimum k value is discussed in Section 5.3.

3.4 Representation of Words with Visterms

To this end we have detected and described keypoints and created a visual vocabulary of keypoints. In order to create the exact representation of word images we use this information in four different ways; histograms of visterms, location weighted histograms of visterms, histograms with soft weighting scheme and string representation of the visterms.

Simply, a word can be represented with the histogram of its visterms, that is the number of each visterm appearing on the word image. This is a pure histogram representation and many studies make use of different representations for term-document relations.



(a)

(b)

Figure 3.4: Patches of the same visterm on **(a)** different instances of a subword from Arabic dataset and **(b)** some words from George Washington dataset. In **(a)** different instances of a subword with letters *ha* and *ya* are seen. Although the writing styles differ, the keypoints shown on the subword instances belong to the same visterm showing that proposed representation is successful on dealing with different writing styles. In **(b)** below part of the letter *t* are defined as the same visterm for different words. Also, connection points of some letters like *m* and *r* refer to same visterm, since it has the same visual patch. This is not exactly what we want from visterm generation, but this shows that our visual vocabulary is visually meaningful. Note that each of the patches are same as the others when go into a convenient affine transformation.

In classical bag-of-features approach, feature vectors are created in *term frequency-inverse document frequency*, which is referred as *tf-idf* format. Tf-idf aims to increase the weight of rarely appearing terms and decrease the weight of mostly appearing terms as used in text retrieval. By this way, if a rarely appearing term exists for an image, this feature is made more representative and distinctive than the others for that image. Our histogram of visterms representation is either in pure histogram form or normalized histogram form or with tf-idf representation.

Since classical histograms do not make use of location information effectively, we also prefer to propose different histogram representations, which are weighted according to locations. Also, Soft Weighting scheme introduced by [21] is used as an alternative way of location usage, where every visterm weights to histogram according to its distance to the visterm centroid.

Since we have a visual codebook, we also create a string representation of the codebook for a word image, by appending the visterms according to writing order. This representation is used for adapting string matching algorithms to our problem later on.

3.4.1 Histogram of Visterms

Mainly, we represent each word image with the histogram of its existing visterms and these histograms are either in normalized form, as we refer $hist_{norm}$ or in *term frequency inverse document frequency* that is *tf-idf* form.

Text retrieval systems mostly use tf-idf representation, where each bin of the histogram is a product of two terms, term frequency and inverse document frequency. For example each document is represented by a k element vector, V_d = (t_1,...,t_i,...,t_k) where term frequency, tf of i^{th} element is

$$tf_i = \frac{n_{id}}{n_d} \tag{3.2}$$

and inverse document frequency, idf of i^{th} element is

$$idf_i = log\frac{N}{n_i} \tag{3.3}$$

and i^{th} term is product of the two

$$t_i = tf_i \cdot idf_i = \frac{n_{id}}{n_d}log\frac{N}{n_i} \tag{3.4}$$

Here n_{id} is the number of occurrences of visterm i in the image d, n_d is the number of visterms in image d, N is the number of images in the whole dataset and n_i is the number of images visterm i appears. At the end we have a N by k, image - visterm matrice in tf-idf form, where k is the number of visterms. Thus we represent the dataset with this matrice. Term frequency increase the weight of a term appearing often in a particular document, and thus describe it well, while idf decrease the weight of a term appearing often in dataset.

A simple illustration of classical histogram creation is shown on Figure 3.5. Figure 3.6 shows three images from Arabic dataset with its affine covariant regions and tf-idf represented feature vector. The city names, that have common parts have histograms resembling each other, while histograms of exactly different words strongly differ.

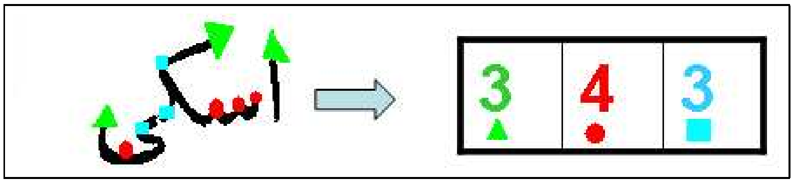
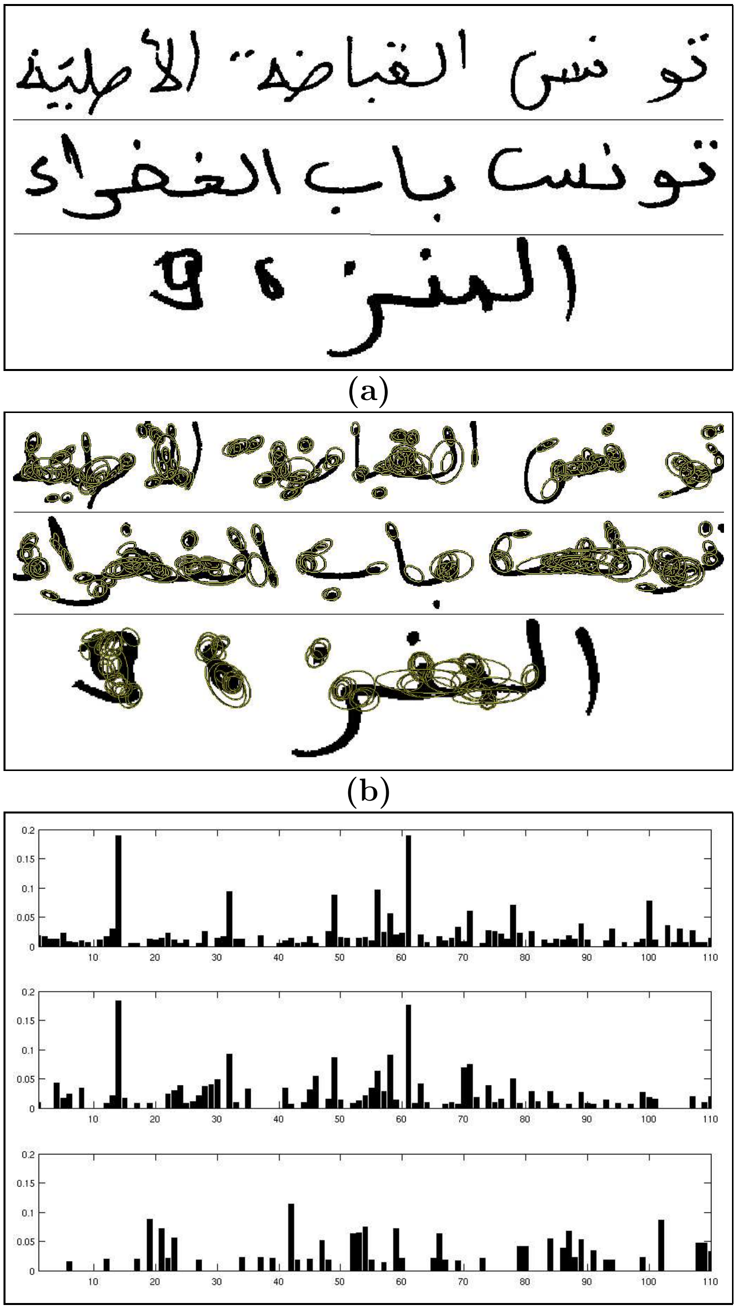


Figure 3.5: Illustration of classical histogram creation for a word image from Ottoman dataset

3.4.2 Location Weighted Histograms

When we represent each word with the histograms, we do not use location information of visterms effectively. However, locations of keypoints give relevant



(a)

(b)

(c)

Figure 3.6: **(a)** Three words from Arabic dataset, **(b)** affine regions of the words and **(c)** *tf-idf* represented feature vectors of the words. Note that first and second words have a common part at the beginning, so that their histogram also resemble each other. On the other hand the third word image is exactly different from the other two. Therefore its representation is not likely the others.

information for the image. For example, if a dot is at the beginning of a word and at the end of another word, these two words must not be same and their representation also must be different. In order to make use of location information, we used

1. Gödel encoding with primes, referred as *Prime*,
2. Encoding with base-2, referred as $Base_2$,
3. Soft Weighting with k neighbors, referred as *SoW-k*.

3.4.2.1 Gödel encoding with primes (***Prime***)

Gödel encoding maps a sequence of numbers to a unique natural number [17]. Suppose we have a sequence of positive integers $s = x_1 x_2 ... x_n$. The Gödel encoding of the sequence is the product of the first n primes raised to their corresponding values in the sequence

$$Encoding(s) = 2^{x_1} \cdot 3^{x_2} \cdot 5^{x_3} ... {p_n}^{x_n} \tag{3.5}$$

We split each word image into fixed size bins, where bin size is 30 pixels, that is average length of a character. This operation is done from right to left for Ottoman and Arabic, since they are written from right to left. Suppose a word has n bins. Then, for each bin of visterm histograms we create a sequence of numbers, with the number of keypoints appearing in the bins of word image from left to right. Thus i^{th} bin of visterm histogram of word W is $W_i = \{s_1, s_2, \ldots, s_n\}$ where s_j is the number of i^{th} visterms appearing in the j^{th} bin of word. Then each sequence is encoded with Gödel encoding with prime numbers. Figure 3.7 illustrates prime-encoding on the visterms of an example word image.

3.4.2.2 Encoding with base-2 ($Base_2$)

Another alternative way of location encoding is to form a digit of a base-K numeral system by mapping each term of sequence to a mapping h. Thus the

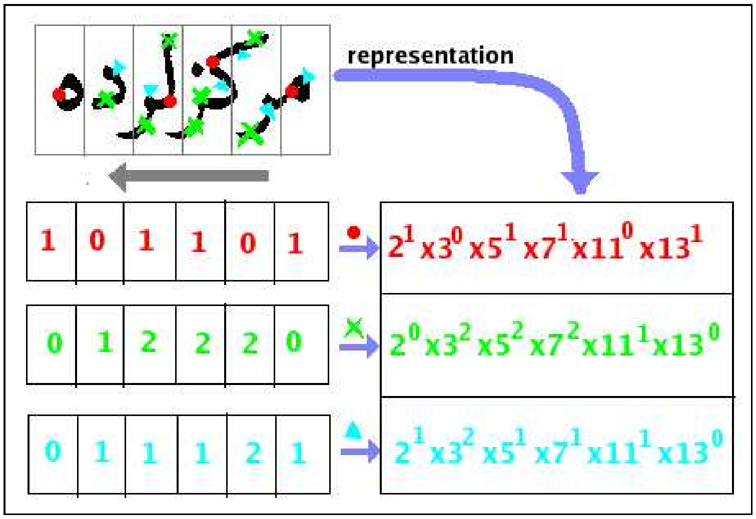

representation
1 0 1 1 0 1
0 1 2 2 2 0
0 1 1 1 2 1
2^1 x3^0 x5^1 x7^1 x11^0 x13^1
2^0 x3^2 x5^2 x7^2 x11^1 x13^0
2^1 x3^2 x5^1 x7^1 x11^1 x13^0

Figure 3.7: Illustration of a prime encoding on an example word from Ottoman dataset. Words are splitted into fixed size bins and each visterm histogram is encoded separately as seen from the figure. The resulting numbers forms the representation for that word image.

sequence above can be encoded as

$$Encoding(s) = h(x_1) \times K^{n-1} + h(x_2) \times K^{n-2} + \ldots + h(x_n) \times K^0 \tag{3.6}$$

We used this encoding scheme similar to *Primes*, where we split each word image into bins and encode the sequence of numbers created for each visterm. We name this representation as $Base_2Bins$ throughout the thesis, since we used sequences coming from the bins.

We also used base-k representation directly for location of keypoints. Suppose we have a keypoint p with location information (x, y), then this keypoint adds 2^x to the respective bin of visterms histogram. Thus, the points $p_1, p_2, \ldots, p_n$ with locations $x_1, x_2, \ldots, x_n$ belonging to the i^{th} visterm, generates i^{th} bin of histogram as

$$H_i = \sum_{j=1}^{n} 2^{x_j} \tag{3.7}$$

This number can also be normalized with the greatest base-2 number created for that sequence, that is

$$H_i = \frac{\sum_{j=1}^{n} 2^{x_j}}{\sum_{j=1}^{n} 2^j} \tag{3.8}$$

Resulting representations are named as $Base_2Locs$ and $Base_2Locs_{norm}$ respectively.

3.4.2.3 Soft Weighting (*SoW*)

Soft-weighting scheme assumes that weighting of a visterm is based on its distance to the cluster centroid. Suppose we have a visual vocabulary of k visterms $V_d = (t_1,...,t_i,...,t_k)$ with each component t_k representing the weight of a visual word k in an image such that

$$t_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(j,k) \tag{3.9}$$

where M_i represents the number of keypoints whose ith nearest neighbor is visterm k and $sim(j,k)$ represents the similarity between keypoint j and visterm k. The contribution of each keypoint is weighted with $\frac{1}{2^{i-1}}$ representing the word is its i^{th} nearest neighbor. This idea weights each keypoint according to its distance to visterm centroid. In the study of Jiang *et al.*[21] N is empirically taken as four. We tried SoW both for one neighborhood and four neighborhood in our study.

3.4.3 String Representation

After visterm generation, we have a vocabulary of k elements and we can rewrite each word image with the help of this visual vocabulary. Since Arabic and Ottoman scripts are written from right to left, we list existing visterms of words from right to left and create a string representation of the visual vocabulary for that word image. This type of representation is used for adaption of string matching algorithms to our problem as told in Chapter 5. Figure 3.8 illustrates string representation of an example word image from Ottoman dataset.

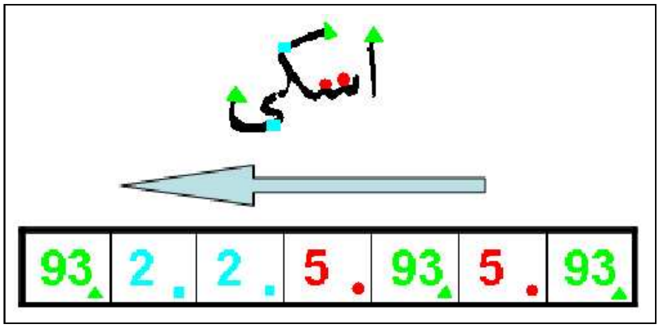


Figure 3.8: An example string representation for a word image from Ottoman dataset. Red points belong to fifth visterm, green points belong to 93rd visterm and blue points refer to second visterm and string creation is done from right to left due to the characteristic of Ottoman script. Note that these are not detected keypoints exactly, this example is only for illustration.

Chapter 4

Retrieval and Indexing

Since our main view is treating words as images rather than sets of characters, important points of word images, that make them distinct from others are detected and described. Then, each word image is identified with the help of the visual vocabulary generated by the vector quantization of keypoint descriptors. This representation is mainly used for retrieval purpose. We also used word spotting idea in order to achive indexing.

This chapter overviews retrieval and indexing steps by introducing distance measures used for retrieval, types of classifiers used for word spotting and pruning step used for elimination.

4.1 Matching for Retrieval and Indexing

In order to make retrieval, each word image is matched with others. Matching aims to find the instances of a query word and a word pair is similar if their feature vectors are also similar. In order to find the similarity between feature vectors, different distance measures are tried as explained in the following section.

4.1.1 Distance Measures used for Matching

In this study each word image is represented either with the classical histogram of its visterm or location weighted histograms. With the fact that the proposed representation is a kind of distribution, we used different types of measures for finding the similarities between word images.

Suppose we have two word images P and Q with corresponding feature vectors $V_p = (p_1, p_2, \ldots, p_n)$ and $V_q = (q_1, q_2, \ldots, q_n)$. Euclidean distance, which is also named as L2 distance, aims to measure the ordinary distance between two spatial points. Euclidean distance between P and Q is calculated according to the formula

$$D_{euclidean} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{4.1}$$

KL-divergence is a measure of the difference between two probability distributions, a true distribution and an observed distribution. KL-divergence distance between an existing distribution P and an observed distribution Q is

$$D_{kl-div} = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \tag{4.2}$$

Since KL-divergence is not symmetric, it is not an exact distance metric. Hence, many studies makes use of symmetric KL-divergence, which is computed with the following formula

$$D_{kl-div} = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} + q_i \log \frac{q_i}{p_i} \tag{4.3}$$

Chi-square metric assumes that two distributions are correspondent, if they populate and grow in the same manner. Chi-square distance between P and Q is

$$D_{chi-sq} = \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{p_i} \tag{4.4}$$

Weighted inner product distance is related to the similarity of the quantity of two variables, while binary inner product only takes their existence into account.

Weighted and binary inner product distance are computed with the following equations

$$D_{weighted} = \sum_{i=1}^{n} p_i \times q_i \tag{4.5}$$

$$D_{binary} = \sum_{i=1}^{n} and(p_i, q_i) \tag{4.6}$$

where $and(p, q)$ is 1 if both p and q are nonzero, 0 otherwise.

Similar to binary inner product, XOR distance computes the number of variables existing and non-existing in two distributions at the same time. XOR distance between P and Q is found as

$$D_{xor} = \sum_{i=1}^{n} xor(p_i, q_i) \tag{4.7}$$

where $xor(p, q)$ is 0 if both are zero or nonzero at the same time and 0 otherwise. Binary inner product and XOR distance deals with the binary representation of histogram. Thus the existence of a visterm is important for these measures rather than the quantity of it. XOR distance counts the number of visterms that exist and do not exist in two images at the same time, while binary inner product only counts the number of visterms that exist in both images.

Cosine distance, which is arc-cosine of the angle between two vectors, aims to find the similarity between directions of the vectors. Cosine distance between P and Q is

$$D_{cosine} = 1 - \frac{\sum_{i=1}^{n} p_i \times q_i}{\left(\sum_{j=1}^{n} {p_j}^2\right)^{1/2} \left(\sum_{j=1}^{n} {q_j}^2\right)^{1/2}} \tag{4.8}$$

During retrieval step, the distance between a query word and the whole set is computed according to above distance measures and ranked based on this similarity. For word spotting, some of the distance measures are tried in order to find the similarity between two samples, but mainly L2 distance is preferred.

4.1.2 Matching with Pruning

In [32] Manmatha *et al.* used a pruning step in order to find the suitable pair for matching step by making use of area and aspect ratio of the word image. They assumed that aspect ratio and area of word images, that are similar, must be close. So they assumed that

$$\frac{1}{\alpha} \geq \frac{AREA_{word}}{AREA_{query}} \geq \alpha \tag{4.9}$$

where $AREA_{query}$ is the area of query word and $AREA_{word}$ is the area of word image to be matched and α values are tried between 1.2 and 1.3 typically through their experiments. A similar pruning is done for aspect ratio (i.e. width/height) of word image and it is assumed that

$$\beta \geq \frac{ASPECT_{word}}{ASPECT_{query}} \geq \beta \tag{4.10}$$

where $ASPECT_{query}$ is the aspect ratio of query word and $ASPECT_{word}$ is the aspect ratio of word image to be matched and different values ranging from 1.4 to 1.7 are tried for β.

For comparison purpose, we also apply the proposed method after the same pruning step on the data, which is also used in [48]. In addition, as we observed that the word images having greater number of keypoints are more representative than the others and large word images have greater number of visterms than the others, this pruning step will help the system to eliminate the words having rather different number of keypoints.

4.2 String Matching

As we generate a visual vocabulary for each dataset, we also represent each word image with the help of this visual vocabulary and named it as string representation. This representation is treated like a kind of string and some experiments are done for matching these representations.

String matching algorithms try to find a place where one or several strings are found within a larger string or text. One of the most popular string matching algorithms is *Minimum Edit Distance Algorithm*, also known as *Levenshtein distance algorithm*, which computes the minimum number of operations to form a string from another one [27].

Algorithm 1 Algorithm for finding min edit distance between two words $s1$ with length m and $s2$ with length n [27]

```
for $i = 1$ to $m$ do
  $D[i+1, 1] \leftarrow D[i, 1] + delCost$
end for
for $j = 1$ to $n$ do
  $D[1, j+1] \leftarrow D[1, j] + insCost$
end for
for $i = 1$ to $m$ do
  for $j = 1$ to $n$ do
    if $s1(i) = s2(j)$ then
      $replace = 0$
    else
      $replace = replaceCost$
    end if
    $D(i+1, j+1) = min(D(i,j) + replace, D(i+1, j) + delCost, D(i, j+1) + insCost)$
  end for
end for
return D(m+1,n+1)
```

The operations to translate a string to other are insertion of an additional character to a specific location of string, deletion of a character from the string and transposition, that is swap of two characters in the string. Levenshtein Distance algorithm is mostly used in natural language processing applications like spell checking and grammar checking. The algorithm dynamically tries to find the minimum number of operations required to form a string from the other as shown in Algorithm 1.

4.3 Indexing

Automatic transcription of handwritten manuscripts can provide strong benefits, but it is still an open problem and fully automatic transcription is nearly impossible. However, the proposed representation is convenient for making recognition on word level. Hence we also perform word spotting by using the proposed representation.

In multi-class classification, there are many objects of different classes and as the number of classes increase, feature space that will cover all objects becomes so difficult to find[57]. In addition, many studies emphasize on detection of only one kind, while other types are usually thought as outliers. So one-class classifiers give more robust solutions for recognition of objects of one class. Based on these properties, we mainly make one-class classification for recognition of words as we used the classifiers listed below.

1. Support Vector Machine
2. K-Nearest Neighborhood
3. Spectral Angle Mapper Classifier
4. Incremental Support Vector Classifier
5. K-means Classifier
6. Parzen Classifier
7. Gaussian Classifier
8. Mixture of Gaussian Classifier

One-class classifiers aims to find a boundary between target and outliers on the feature space. Support Vector Data Description (SVDD) tries to draw a hypersphere around target objects and the volume of this hypersphere is minimized in order to minimize the chance of accepting outliers. It maps the data to a new, high dimensional feature space without much extra computational costs and solves the problem in this space. Similarly incremental SVDD applies the same algorithm as in SVDD, this time for huge amount of data and some restrictions on space and time.

K-Nearest neighborhood classification assigns each test object to the mostly appearing class among its k nearest neighbors. K-means classifier learns the centroids of each class from training samples and assigns the coming test samples to the class of nearest centroid. As the name implies, centroids are chosen as the mean of training samples in the class.

Parzen classifier estimates the threshold of probabilities with observed data and assigns each test sample to a class according to this threshold value. Similarly Gaussian classifiers estimates a Gaussian distribution from the test samples and decide whether coming test sample belong to this distribution or not. Mixture of Gaussian classifier applies the same step as in Gaussian classifier with sum of multiple Gaussian distributions.

Spectral Angle Mapper Classifier (SAMC) behaves each sample as a vector and computes a spectral angle for the classes and then assigns the test sample to the class having minimum angle difference with questioned sample. Thus SAMC base on the direction of feature vector rather than magnitude of it.

In [25], Lavrenko *et al.* generate a HMM classifier for word recognition on a historical documents set, which we also used for word recognition by using the proposed representation. Rather than generating a complicated HMM classifier, we make classification by using the classifiers listed above. Recognition is also done for Arabic dataset, which is used in many studies for recognition purpose and comparative results are given in Chapter 5.

Chapter 5

Experiments and Results

In this chapter, first, characteristics of the datasets used in this study will be explained. Second, word segmentation process, which is performed before matching, is explained briefly and a method for optimizing the k value used in visterm generation will be discussed. Finally, retrieval and indexing results will be reported and evaluated in related sections.

5.1 Datasets

Initially, we started by aiming to solve the problem for Ottoman documents, but since accessing Ottoman manuscripts was difficult for us as to many researchers and there does not exist Ottoman dataset with ground truth information, we prefered to perform some additional comparative experiments on a large Arabic dataset with a convenient ground truth information, with the observation that Ottoman and Arabic scripts particulary resemble each other.

There are numerous historically invaluable text documents in all languages. Among one of them, George Washington collection of Library of Congress [28] contains a large collection of historical manuscripts and correspondences. Since we aim to propose a generalized solution for historical documents, we also ran experiments on a dataset generated from George Washington collection, since it has been previously experimented on with word matching techniques. Writing style and the alphabet of this set is different from Ottoman and Arabic. So we also carried out experiments for finding the suitable vocabulary size and representation type for this dataset. Results of the experiments on this set shows that the performance of the proposed method is comparable with the most recent methods on this field.

Consequently, we studied on three different scripting styles that are Ottoman, Arabic and handwritten Latin. In the upcoming sections, detailed information about scripting styles and datasets will be given.

5.1.1 Ottoman Datasets

Ottoman archives, being one of the largest collections of historical documents, hold over 150 million documents ranging from military reports to economical and political correspondences belonging to the Ottoman era [58]. A significant number of researchers from all around the world are interested in accessing the archived material [41]. However, many documents are in poor condition due to age or are recorded in manuscript format. Thus, manual transcription and indexing of Ottoman texts require a lot of time and effort, causing most of these documents to become inaccessible to public research.

Ottoman script is a connected script based on Arabic alphabet with additional vocals and characters from Persian and Turkish languages [18] and therefore shares the difficulties faced in Arabic.

During Ottoman imperial era, calligraphy was a respected and highly regarded art such that different calligraphy styles were created and used by Turks in the history [59]. Calligraphy styles and some Tuğra (signature of Sultan) examples are shown in Figure 5.1. As seen from the examples splitting characters from Ottoman documents is a challenging problem.

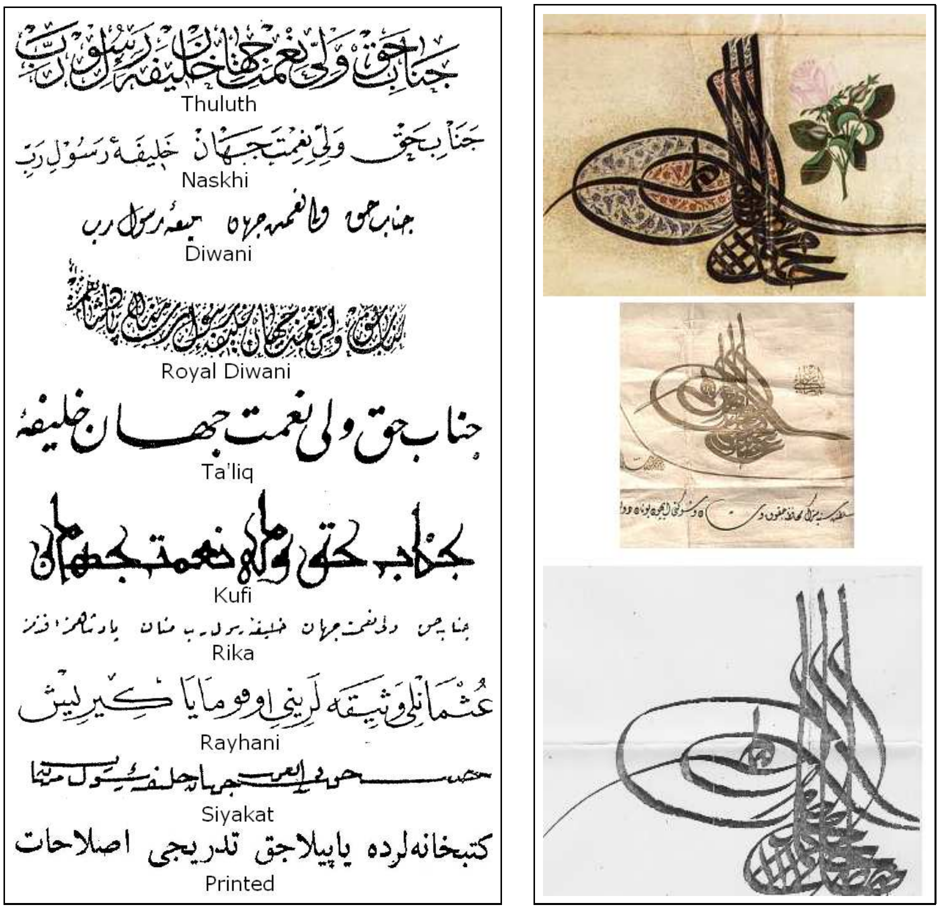
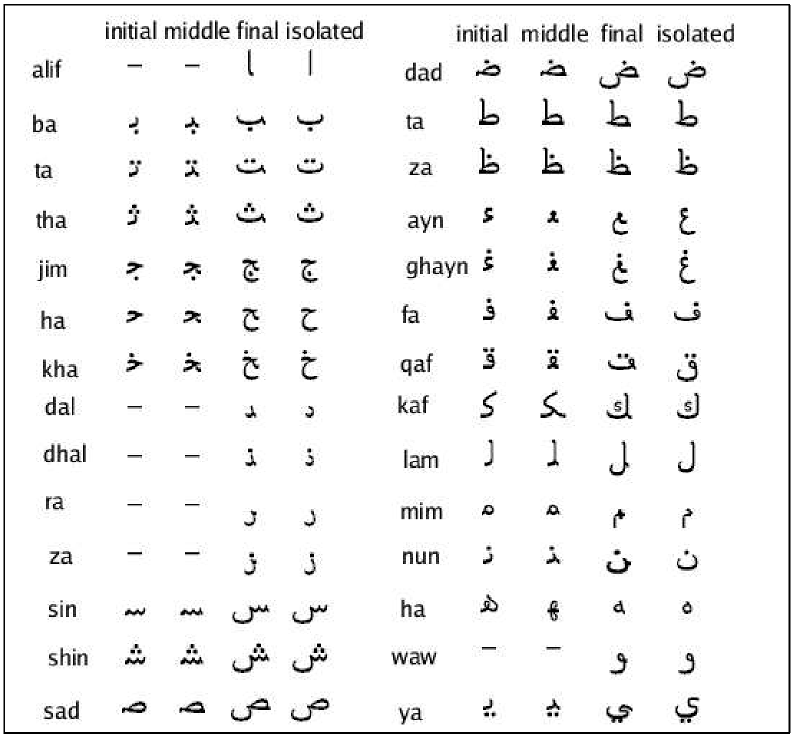


Figure 5.1: **on the left** Calligraphy styles used during Ottoman era [15], **on the right** some tuğra (signature of Sultan) examples. In Ottoman Empire every Sultan has a particular signature, which is named as tuğra and these signatures are drawn on the governmental documents by some special people under the rule of the Sultan.



	initial	middle	final	isolated		initial	middle	final	isolated
alif	–	–	ـا	ا	dad	ضـ	ـضـ	ـض	ض
ba	بـ	ـبـ	ـب	ب	ta	طـ	ـطـ	ـط	ط
ta	تـ	ـتـ	ـت	ت	za	ظـ	ـظـ	ـظ	ظ
tha	ثـ	ـثـ	ـث	ث	ayn	عـ	ـعـ	ـع	ع
jim	جـ	ـجـ	ـج	ج	ghayn	غـ	ـغـ	ـغ	غ
ha	حـ	ـحـ	ـح	ح	fa	فـ	ـفـ	ـف	ف
kha	خـ	ـخـ	ـخ	خ	qaf	قـ	ـقـ	ـق	ق
dal	–	–	ـد	د	kaf	كـ	ـكـ	ـك	ك
dhal	–	–	ـذ	ذ	lam	لـ	ـلـ	ـل	ل
ra	–	–	ـر	ر	mim	مـ	ـمـ	ـم	م
za	–	–	ـز	ز	nun	نـ	ـنـ	ـن	ن
sin	سـ	ـسـ	ـس	س	ha	هـ	ـهـ	ـه	ه
shin	شـ	ـشـ	ـش	ش	waw	–	–	ـو	و
sad	صـ	ـصـ	ـص	ص	ya	يـ	ـيـ	ـي	ي

Figure 5.2: Forms of letters in Arabic alphabet for printed writing style. The letters can have four different forms according to its position in the word, initial, middle, final and isolated. Note that some letters like *alif, dal, dhal* cannot be connected to the characters after it. Thus a word can also have small gaps in it.

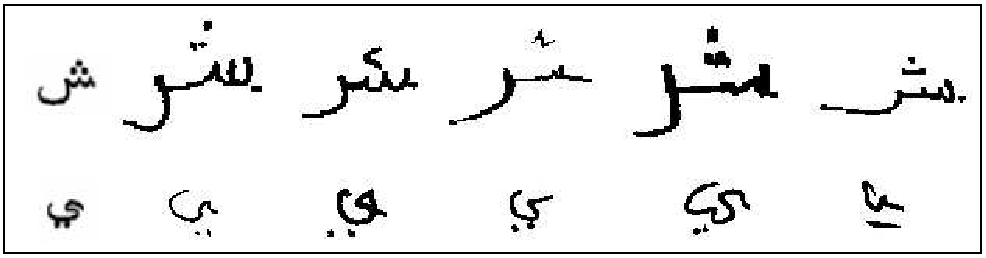


Figure 5.3: Different handwritten forms of letters *shin*(**first line**) and *ya*(**second line**). For each line first letter is in printed form and the remaining are in handwritten form written by different writers. Note that, although the position of points and shape of curves change, a novice person can see them as images resembling each other.

Similar to Arabic, in Ottoman scripts, every character can have up to four alternative forms depending on the relative positioning of the character inside the word (beginning, middle, end and isolated). Figure 5.2 shows different forms of the letters of Arabic alphabet, which are also used in Ottoman alphabet, in printed type writing style. However, if the documents are handwritten, the forms of letters deviate significantly, resulting in complications in recognition of characters by mere eye. As seen in Figure 5.3, a character can be hand written in alternative forms by different writers, since the position of dots and shapes vary. Thus each letter is not essentially in a fixed form, therefore, the shape of it depends on the characteristic style of the writer. In addition, sometimes the styles of the characters may also differ for the same writer.

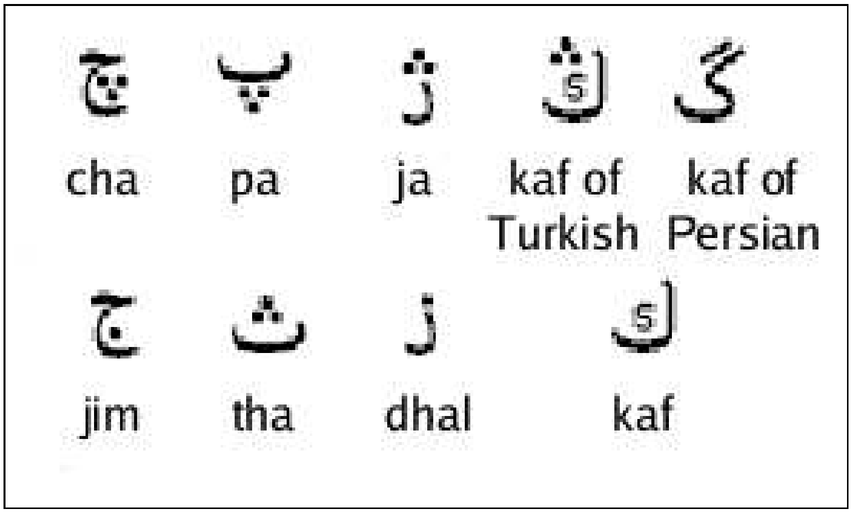


Figure 5.4: Five additional characters existing in Ottoman Alphabet shown in the first line and below are the resembling letters. Note that the letter *cha* resembles the writing of *ha, kha* in addition to *jim*. Also the letters *ba, ta, tha* all resemble *pa*. *Kaf of Turkish* and *kaf of Persian* are different forms of letter *kaf* and are used for expressing some Turkish based vocals especially at the end of words.

The additional letters of Ottoman script are coming from the need of expressing some vocals in Turkish language. As seen in Figure 5.4 these characters are highly similar to some already present others, making recognition of Ottoman characters further challenging.

In both Ottoman and Arabic alphabets, a character can be written in various forms. Sometimes two characters may coincide, making character extraction more difficult. Figure 5.5 shows some examples of difficulties in extraction. Notice that the characters are different from the table of printed characters seen on Figure 5.2.

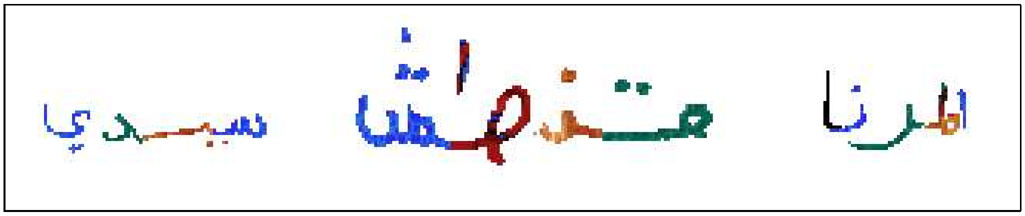


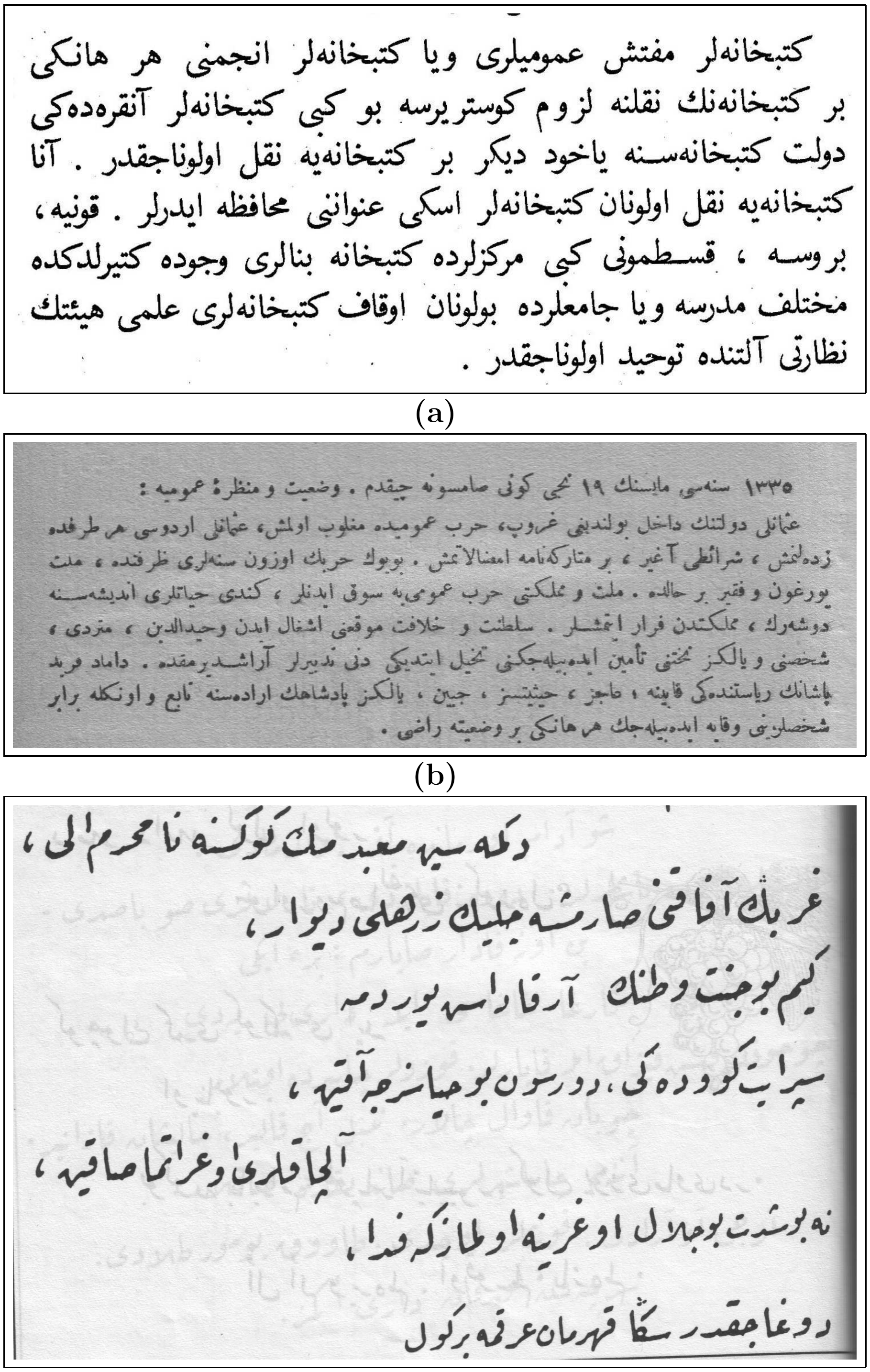
Figure 5.5: Some examples of difficulties of character extraction. Each character is shown with different colors. Note that the rightmost letter is *alif* very closer to the consecutive letters. Following letters are *lam* and *mim* very different from the reported formats in printed writing style when connected.

Another characteristic of Ottoman is that, there are only a few vowels. Therefore, transcription of a word strongly depends on the context of the document and vocabulary of the reader. Sometimes two different words can be written exactly the same, but suitable word is selected according to the context of the document.

There are three kind of document sets for Ottoman script used in this study (Figure 5.6). One of them is in rika writing style, which is mostly used as a handwriting style through governmental correspondences of Ottoman Empire. The others are in printed documents, which is more regular and justified than rika style. We used printed type documents since it is easier to manually annotate them. In addition, accessing manuscripts of Ottoman archives is not an easy task and acquired manuscripts do not have a convenient ground truth information.

The document sets for Ottoman script are small-printed, large-printed and rika.

- **Small-printed:** Manually annotated printed (matbû) documents that takes 823 words in 6 pages [24]. These documents are governmental correspondences about the arrangements in national libraries in the early stages of Turkish Republic. Since the documents have a common topic, they have



کتبخانه‌لر مفتش عمومیلری ویا کتبخانه‌لر انجمنی هر هانکی
بر کتبخانه‌نك نقلنه لزوم کوستریرسه بو کبی کتبخانه‌لر آنقره‌ده‌کی
دولت کتبخانه‌سنه یاخود دیکر بر کتبخانه‌یه نقل اولوناجقدر . آنا
کتبخانه‌یه نقل اولونان کتبخانه‌لر اسکی عنواننی محافظه ایدرلر . قونیه ،
بروسه ، قسطمونی کبی مرکزلرده کتبخانه بنالری وجوده کتیرلدکده
مختلف مدرسه ویا جامعلرده بولونان اوقاف کتبخانه‌لری علمی هیئتك
نظارتی آلتنده توحید اولوناجقدر .

(a)

١٣٣٥ سنه‌سی مایسنك ١٩ نجی کونی صامسونه چیقدم . وضعیت و منظرهٔ عمومیه :
عثمانلی دولتنك داخل بولندیغی غروپ، حرب عمومیده مغلوب اولمش، عثمانلی اردوسی هر طرفده
زده‌لنمش ، شرائطی آغیر ، بر متارکه‌نامه امضالانمش . بویوك حربك اوزون سنه‌لری ظرفنده ، ملت
یورغون و فقیر بر حالده . ملت و مملکتی حرب عمومی‌یه سوق ایدنلر ، کندی حیاتلری اندیشه‌سنه
دوشه‌رك ، مملکتدن فرار ایتمشلر . سلطنت و خلافت موقعنی اشغال ایدن وحیدالدین ، متردی ،
شخصنی و یالکز تختنی تأمین ایده‌بیله‌جکنی تخیل ایتدیکی دنی تدبیرلر آراشدیرمقده . داماد فرید
پاشانك ریاستنده‌کی قابینه ، عاجز ، حیثیتسز ، جبین ، یالکز پادشاهك اراده‌سنه تابع و اونکله برابر
شخصلرینی وقایه ایده‌بیله‌جك هر هانکی بر وضعیته راضی .

(b)

دکمه سینه معبدملك کوکسنه نا محرم الی ،
غربك آفاقنی صارمشه چلیك زرهلی دیوار ،
کیم بو جنت وطنك آرقاداشه یوردمه
سیرایت کوودەکی ، دورسون بو حیاسزجه آقینه ،
آلچاقلری وغراتما صاقینه ،
نه بو شدت بو جلال اوغرینه اولمازکه فدا ،
دوغاجقدر سکا قهرمان عرقمه برکول

(c)

Figure 5.6: Example documents for Ottoman dataset: **(a)** small-printed, **(b)** large-printed, **(c)** rika

many common words. They are automatically segmented into words as told in Section 5.2.

- **Large-printed:** 25 pages of the book Nutuk by Mustafa Kemal Atatürk, which is in the form of printed style [9]. These pages are carefully scanned and automatically segmented to 9524 words after binarization with Otsu method[42]. This set is not manually annotated due to its large size.
- **Rika:** Two pages from Turkish National Anthem, which is written with rika handwriting style, that is less regular and more complicated than printed writing style [15]. The pages are segmented into 257 word images, that are also manually annotated later. Segmentation is done semi-automatically because of the writing style of script, that is lines are segmented automatically and words are segmented manually.

5.1.2 IFN-ENIT Dataset

IFN-ENIT dataset, which is the set of Tunisian city names, is used as a larger set with ground truth to test the performance on Arabic words[45]. It includes 937 Tunisian city names from 411 writers and about 26000 word images divided into four subsets. Many studies make recognition on this set [12, 46], which we refer as *IFN-ENIT-all*. So we also tested the recognition performance of our system on this entire set. In addition we also create another homogeneous subset including 200 unique words with 30 instances each, in order to observe the retrieval performance of our system and referred as *IFN-ENIT-subset* throughout the thesis. Each word-image in the IFN-ENIT dataset is annotated with the zip code of the city, which makes it easier to process for performance analysis. Figure 5.7 shows some example word images from IFN-ENIT dataset.

5.1.3 George Washington Dataset

George Washington dataset is composed of two sets that are taken from George Washington collection of Library of Congress which is made available by Center

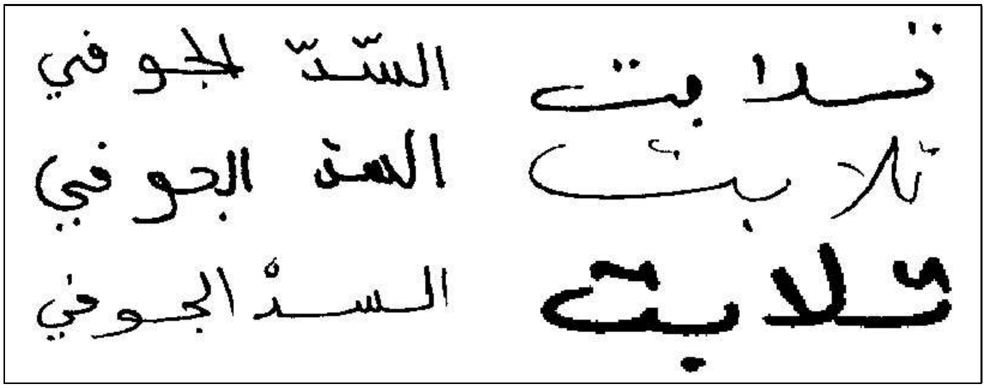


Figure 5.7: 3 instances of two city names *Talabat* **(right)** and *Sadd'el Jufi* **(left)** from IFN-ENIT Dataset. Since the writers are different, word instances differ from each other, but if we observe the words like images, we can say that they are images resembling to each other. Note that some of them written thinner than the others showing that the features based on number of black pixels does not work well.

for Intelligent Information Retrieval of UMASS. This collection is composed of historical handwritten letters written by George Washington with a connected scripting style. The first one is 20 pages long and manually segmented and annotated into 4861 word images [25]. Second one has automatically segmented and manually annotated 2381 word images with a better quality than the first one and it is used in [48] and [20]. These sets are referred as *GW-1* and *GW-2* respectively throughout the thesis. An example page from GW-1 is seen on Figure 5.9.

For collection of handwritten manuscripts belonging to a single author, the images of multiple instances of the same word are likely to look similar, but since they are handwritten, there is a variation in the way the words are written (i.e. slant, skew angles, word lengths.). Also, historical documents are often in poor quality, making recognition harder. Figure 5.8 shows multiple instances of the word *they* throughout GW dataset. Notice that Optical Character Recognition systems do not work since it is a connected writing style and the documents are in poor quality.

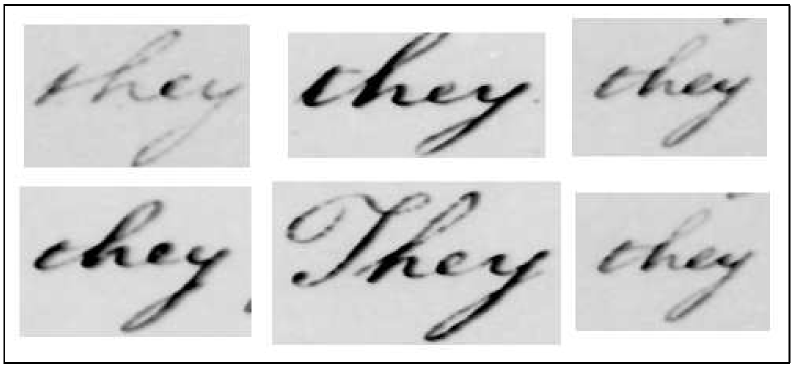


Figure 5.8: Some instances of the word *they*. Note that since the documents are degraded, some instances differ from others.

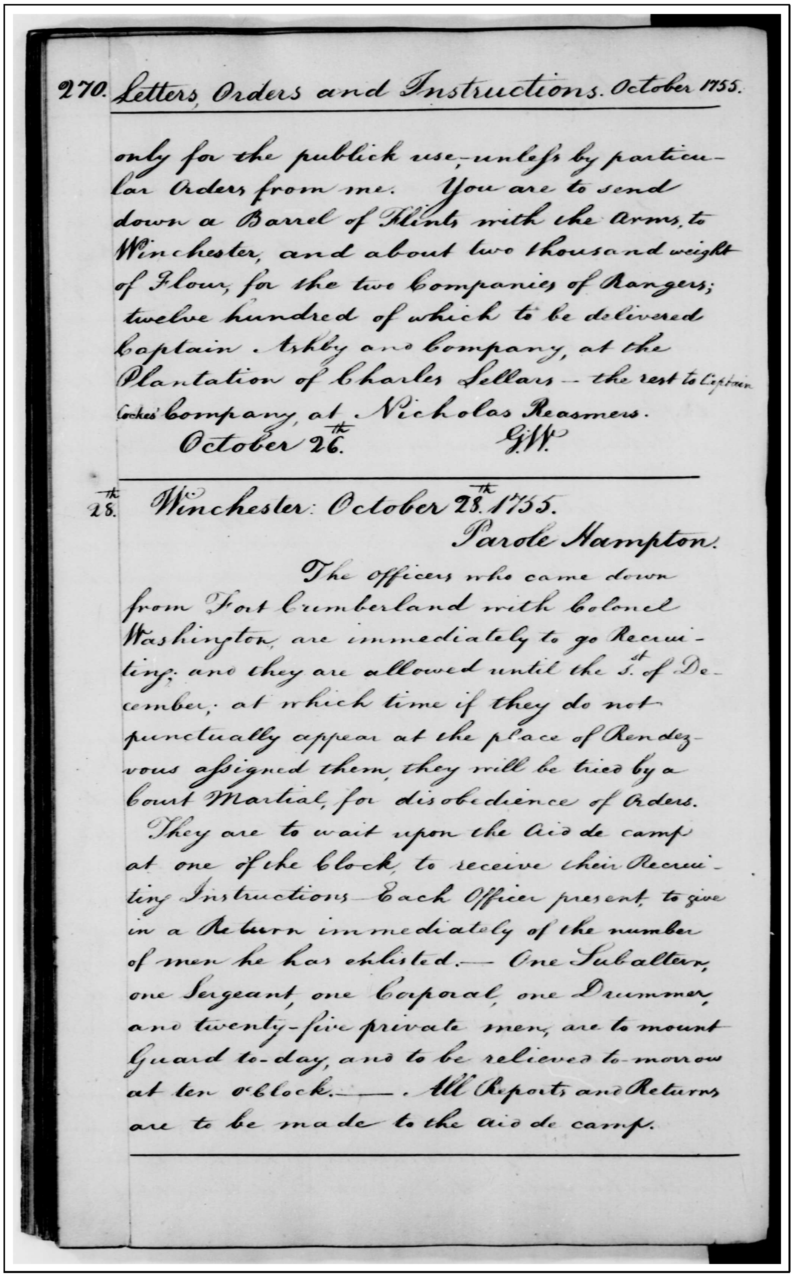
5.2 Segmentation

Since we used a word image matching technique, we need to segment the documents into words before matching step. As listed before GW and IFN-ENIT datasets are already segmented into words and ready for matching, while a segmentation step is required for Ottoman dataset.

For extracting words from documents, simple and commonly used techniques are adopted. First, documents are binarized using the OTSU method [42]. Then, lines and words are segmented using smoothed horizontal and vertical projection profiles respectively. Since documents are carefully scanned, and the writing styles used are mostly written on straight lines, rectification is not required.

Minima on vertical projection profile of a line segment refers to the gaps between words. Horizontal projection profile of an example document and a line segment with its extracted words are shown in Figure 5.10.

With the evaluations on small-printed data set, 100% line segmentation and 82% word segmentation performances are obtained. Figure 5.11 shows some errors in word segmentation. Only on rika data set, after automatic binarization and line segmentation steps, manual word segmentation is performed due to the difficulty of the data set. Segmentation results for large-printed dataset can not



270. Letters, Orders and Instructions. October 1755.

only for the publick use,—unleſs by particular Orders from me. You are to send down a Barrel of Flints with the Arms, to Winchester, and about two thousand weight of Flour, for the two Companies of Rangers; twelve hundred of which to be delivered Captain Ashby and Company, at the Plantation of Charles Sellars—the rest to Captain Cockes' Company, at Nicholas Reasmers.

October 26th. G.W.

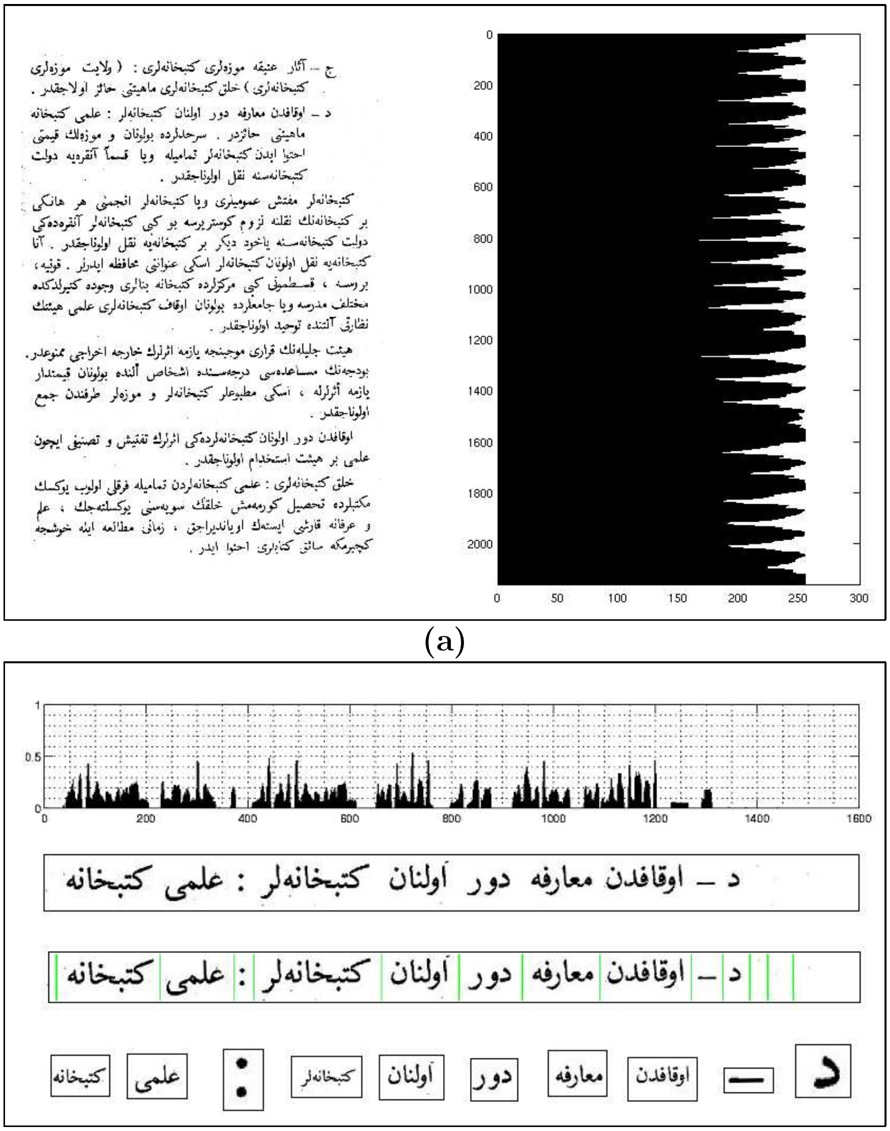
28th. Winchester: October 28th. 1755.

Parole Hampton.

The officers who came down from Fort Cumberland with Colonel Washington, are immediately to go Recruiting; and they are allowed until the 1st. of December; at which time if they do not punctually appear at the place of Rendezvous aſsigned them, they will be tried by a Court Martial, for disobedience of Orders.

They are to wait upon the Aid de camp at one of the Clock, to receive their Recruiting Instructions—Each Officer present, to give in a Return immediately of the number of men he has enlisted.— One Subaltern, one Sergeant, one Corporal, one Drummer, and twenty-five private men, are to mount Guard to-day, and to be relieved to-morrow at ten o'Clock.——. All Reports and Returns are to be made to the Aid de camp.

Figure 5.9: An example page from GW-1.



ج – آثار عتيقه موزه‌لری كتبخانه‌لری : (ولايت موزه‌لری
كتبخانه‌لری) خلق كتبخانه‌لری ماهيتنی حائز اولاجقدر .
د – اوقافدن معارفه دور اولنان كتبخانه‌لر : علمی كتبخانه
ماهيتنی حائزدر . سرحدلرده بولونان و موزه‌لك قيمتی
احتوا ايدن كتبخانه‌لر تماميله ويا قسماً آنقره‌يه دولت
كتبخانه‌سنه نقل اولوناجقدر .

كتبخانه‌لر مفتش عموميلری ويا كتبخانه‌لر انجمنی هر هانكی
بر كتبخانه‌نك نقلنه لزوم كوستريرسه بو كبی كتبخانه‌لر آنقره‌ده‌كی
دولت كتبخانه‌سنه ياخود ديكر بر كتبخانه‌يه نقل اولوناجقدر . آنا
كتبخانه‌يه نقل اولونان كتبخانه‌لر اسكی عنوانلرينی محافظه ايدرلر . قونيه ،
بروسه ، قسطمونی كبی مركزلرده كتبخانه بنالری وجوده كتيرلدكده
مختلف مدرسه ويا جامعلرده بولونان اوقاف كتبخانه‌لری علمی هيئتك
نظارتی آلتنده توحيد اولوناجقدر .

هيئت جليله‌نك قراری موجبنجه يازمه اثرلرك خارجه اخراجی ممنوعدر.
بودجه‌نك مساعده‌سی درجه‌سنده اشخاص آلنده بولونان قيمتدار
يازمه أثرلرله ، اسكی مطبوعلر كتبخانه‌لر و موزه‌لر طرفندن جمع
اولوناجقدر .

اوقافدن دور اولونان كتبخانه‌لرده‌كی اثرلرك تفتيش و تصنيفی ايچون
علمی بر هيئت استخدام اولوناجقدر .

خلق كتبخانه‌لری : علمی كتبخانه‌لردن تماميله فرقلی اولوب يوكسك
مكتبلرده تحصيل كورمه‌مش خلقك سويه‌سنی يوكسلته‌جك ، علم
و عرفانه قارشی ايسته‌ك اويانديراجق ، زمانی مطالعه ايله خوشجه
كچيرمكه سائق كتابلری احتوا ايدر .

(a)

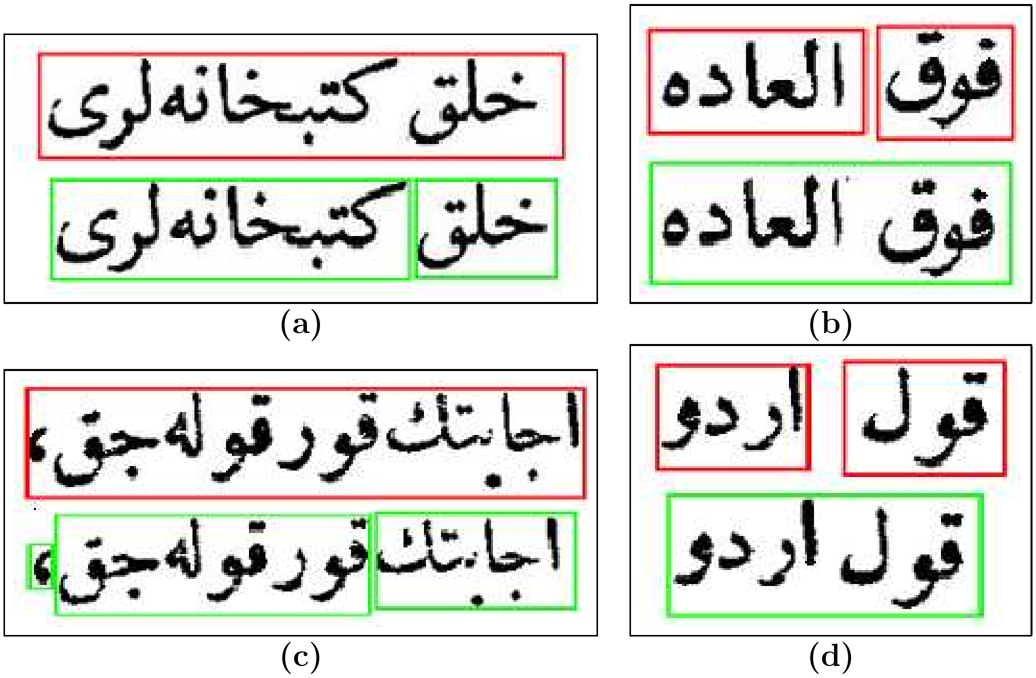
د – اوقافدن معارفه دور أولنان كتبخانه‌لر : علمی كتبخانه

د | – | اوقافدن | معارفه | دور | أولنان | كتبخانه‌لر | : | علمی | كتبخانه

(b)

Figure 5.10: **(a)**Horizontal projection profile of a document, **(b)**Word Extraction: First image is vertical projection profile of the line shown in second image. Third is the split points for that line. Last row is the word images that are extracted from that line.

be given numerically due to its massive size.


خلق کتبخانه‌لری
خلق کتبخانه‌لری
فوق العاده
فوق العاده
اجابتك قورقوله‌جق،
اجابتك قورقوله‌جق،
قول اردو
قول اردو

(a) (b) (c) (d)

Figure 5.11: Some examples of word extraction errors. Red boxes are the wrong extractions and green boxes are the correct extractions. One reason for the segmentation errors is the phrases with very small gaps in between the words. This situation is seen in the phrase at part (a), meaning *Public Libraries*, which could not be split into two words because of the tails of the letters near spaces. Similarly, the words in part (c) also remain unextracted due to elongated letters. On the other hand, isolated format of some consecutive characters may result in large gaps causing over-segmentation as seen in the words at (b) and (d).

One should note that better methods could be applied for preprocessing, but our focus is on representation of words after segmentation. Therefore, in this stage, we chose the simplest methods with the knowledge that better segmentation would result in better retrieval and indexing performance.

We should also mention that word segmentation errors could be tolerated with the proposed approach. For example, if a single word is incorrectly segmented into two parts, it is likely that the subparts will be matched with the original word with relatively large scores since the proposed approach is able to capture the semantic relations between words which have common parts.

5.3 How to find the best k value for visterm generation

Characters in Ottoman alphabet may have four different forms according to the position in the word (initial, middle, final, and isolated) as told in previous section. Usually, these different forms of the same character are very similar. Also, although some characters may include some common parts, the similarity among these different forms is higher than the similarity to other characters.

Based on these observations, we force the keypoints in different forms of the same character to be in the same cluster and the keypoints of different characters to be in separate clusters, by computing an error measure for different k values in k-means while obtaining the visterms.

For this purpose, a codebook of 117 elements which includes up to four different forms of 31 characters in the alphabet is created by also considering different connections to other characters. Note that in Turkish, some characters may not be at the initial or final position of a word. This is why the codebook includes 117 elements rather than 124 elements. In addition, some letters like the forms of letter *kaf* only exist at the end of words.

Created codebook is then used to choose the best number of clusters, k, in k-means by minimizing the following error:

$$error = 1/C \sum_{i=1}^{C} c_i + 1/M \sum_{j=1}^{M} m_j \tag{5.1}$$

Here C is the number of characters in the alphabet, c_i is the number of clusters that a character c appears, M is the number of clusters, and m_j is the number of different characters that the cluster j includes. The error measure corresponds to the sum of the average number of clusters for a character with the average number of characters for a cluster. We assume that a character is in a cluster if a keypoint belonging to any form of that character is in that cluster.

Using 1256 keypoints from the codebook, by incrementing k values by 10 between 20 and 200, the minimum error is achieved for $k = 110$.

In order to test the effectiveness of this method, we use the clusters obtained by the codebook and perform queries on each element in the codebook using the proposed method. Figure 5.12 shows two examples of character queries. As can be seen, different forms of the same character can be successfully matched, and the wrong matches are usually due to similar sub-patterns.

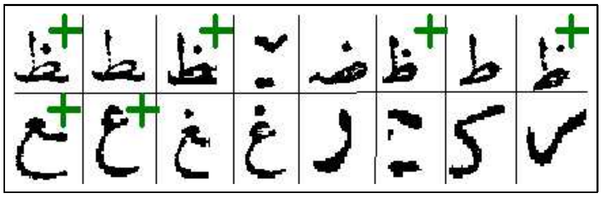


Figure 5.12: Example query results on codebook. .

Although, this codebook could also be used for initializing the k-means, since the results were not very different than using random initialization, we prefer not to use the codebook further. In the experiments, only k=110 value is used to set the number of clusters, but the clusters are randomly initialized using the elements of the working data set.

5.4 Retrieval Results

As we handle the word recognition problem like a novice user, we see the words as images and try to make image matching based on our observations. We use connection and curvature points and dots as distinctive features for comparing and distinguishing two images. Ottoman documents contain image like regions, like miniatures, signatures and illuminations.

In Ottoman Empire, every Sultan had a signature, which is named as Tuğra and Tuğras resemble drawing rather than writings. Hence, in order to test the performance of our approach, we also ran our system on 6 Tuğra images, where three of them belongs to Sultan Ahmet III and the others belong to Sultan Abdülhamit II. As seen in Figure 5.13 proposed method is able to find similar signatures showing that approaching the problem in a word-image matching bases gives a good solution. As seen from the figure, Tuğras strongly show the figure-like nature of Ottoman calligraphy and thus extracting characters from the Tuğras is nearly impossible, therefore character based techniques are unable to solve the problem.

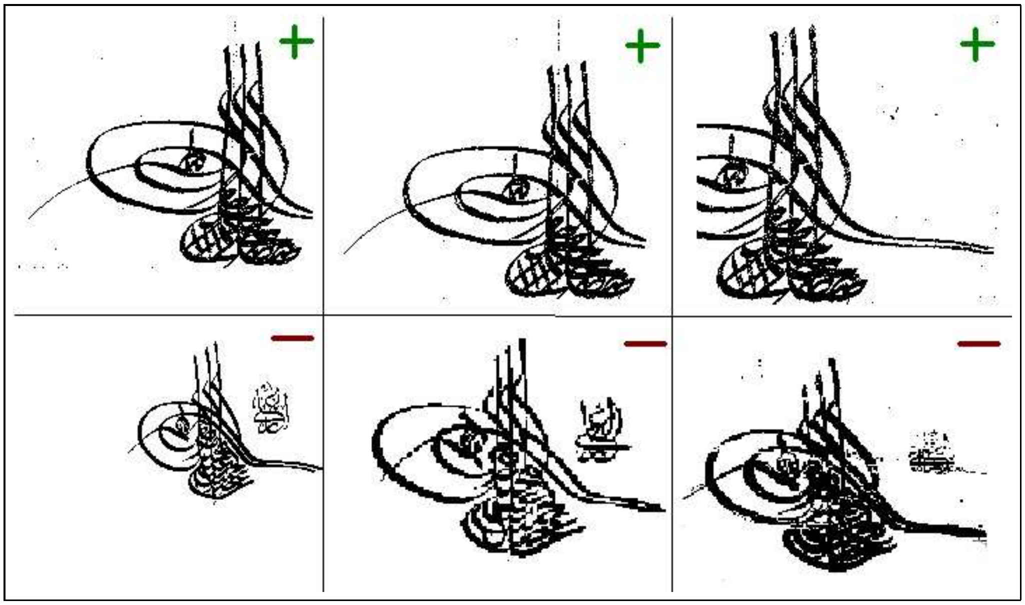


Figure 5.13: Retrieval results for tuğras (Sultan signatures). Top left one is queried among the set and ranked results are displayed. Correct matches are indicated with + and false matches are indicated with - signs. Three instances of the query tuğra are retrieved in the top three ranks. Notice that, character extraction from tugras seems nearly impossible. Note also that, although the instances of the query tuğra has some missing parts, the proposed method can successfully retrieve them in the top ranks.

Experiments are done on all datasets separately and for computation of performance, each word image is queried in the dataset and mean Average Precision (mAP) value is calculated among all queries. Precision is the fraction of retrieved images, that are relevant. For a query, average precision is average of the precisions, that are calculated after a relevant item is retrieved. Suppose we have n instances of a query image and after querying i^{th} relevant image is retrieved at $rank_i$. Thus average precision for that query is

$$\sum_{i=1}^{n}(\frac{\#relevant}{\#retrieved})_{rank_i} \tag{5.2}$$

where $(\frac{\#relevant}{\#retrieved})_{rank_i}$ is the precision value when i^{th} relevant is retrieved.

All techniques are run on IFN-ENIT-subset, since it is larger and more difficult with different writers and writing styles. George Washington dataset is used for mainly comparing our method with previous methodologies. Based on the comparative results on IFN-ENIT-subset, the most suitable method is applied to Ottoman dataset.

5.4.1 Retrieval Results for IFN-ENIT-subset

We have carried out experiments on IFN-ENIT-subset with different representation of feature vector and different distance measures. Table 5.1 shows results on classical histogram represented feature vectors and Table 5.2 shows results based on location weighted histograms (LWH).

In Table 5.1, performance for different detector and representation types are presented. As seen from the results, tf-idf represented feature vectors give equal or even better performance compared to the other. Because, tf-idf increases the weight of more distinctive visterms and decreases the importance of frequently appearing visterms that are treated as stop words.

The results also show that, detecting points with Harris Affine detector gives better results than DoG detector. That is because, affine transformation may not be equal in each direction and handwriting is closer to affine transformations rather than scale changes as in DoG detector. Note that when we used Harris Affine detector, we used additional 5 more parameters that represent affine transformation, in the description of keypoint. These parameters also include location information of keypoint, thus we make use of location information in particular.

Among the distance measures, generally cosine distance is better than the others. We can also see that Euclidean distance, which is mostly used in many

Table 5.1: Results on IFN-ENIT-subset with $hist_{norm}$ and *tf-idf* represented feature vectors using Harris Affine and DoG detector. Note that, the best results are achieved for Harris Affine detector and tf-idf representation, hence we prefer to use them for further studies.

	Harris Affine		DoG
Distance Type	$hist_{norm}$	tf-idf	tf-idf
Cosine	.0764	.0757	.0526
KL-Divergence	.0569	.0674	.0600
Chi-Square	.0579	.0573	.0448
Euclidean	.0671	.0664	.0495
Weighted Inner Product	.0333	.0603	.0394
Binary Inner Product	.0493	.0493	.0452
XOR	.0772	.0772	.0607

Table 5.2: Results on IFN-ENIT-subset with Location Weighted Histograms and Soft Weighting (SoW) scheme. Note that after evaluating the results, we prefer to use tf-idf represented feature vectors.

Prime	$Base_2$Bins		$Base_2$Locs		SoW	
	Norm.	Not Norm.	Norm.	Not Norm.	1 neigh.	4 neigh.
.0444	.0510	.0524	.0377	.0718	.0765	.0758

applications, is not so successful for finding similarity between distributions. Statistical distance measures like KL-divergence performs better than Euclidean distance for distribution matching. XOR distance, which is a binary measure counting the existence and nonexistence of visterms also results in a good performance. This shows that existence or nonexistence of a visterm in a word image is a more representative information than number of visterms appearing in the word image.

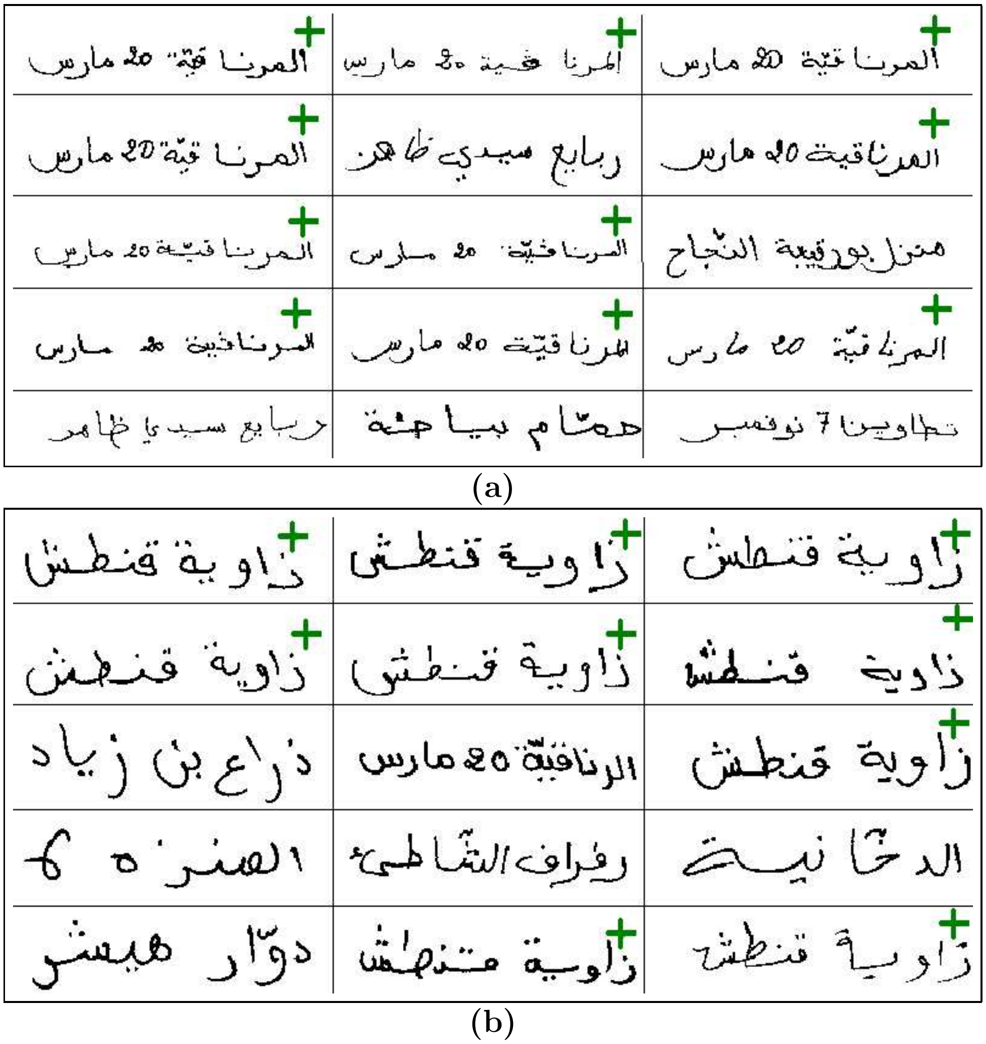
As explained before, classical histograms does not use location information directly, thus we used LWHs. Table 5.2 shows results for LWH representation where keypoints are detected with Harris Affine detector. As can be seen from the table, Gödel encoding does not give good performance, because slight length difference between words can result in big difference in encoding, since encoding is done with primes and base-2 numbers. For example two similar words can have 15 and 16 bins respectively, but this difference weights encoding with 2^{16} or power of 16^{th} prime number. Causing from similar reasons, $Base_2Locs$ feature vector, which i^{th} visterm is computed with the sum of 2^x of keypoints where x is the location information of related keypoint gives better result than $Base_2Bins$. This is also not unexpected, since it directly uses location information rather than splitting the words into bins.

Another location dependent representation, SoW scheme, shows a better performance among the table, where there is only slight difference between one neighbor and four neighbors SoW. This shows that representing each keypoint with only one cluster, is a convenient way for retrieval and visterm generation is successful on classifying patches.

Although SoW scheme shows the most successful performance among the others in Table 5.2, it is approximately same with the best results of unweighted histograms as reported in Table 5.1. Hence, we prefer to use Harris Affine detector with *tf-idf* representation and matching with cosine distance. Hence, comparison with other methodologies will be done by using this representation and measure type, where we referred as *proposed method* throughout the thesis. Selected representation gives accurate results for the retrieval of documents. Some query results with the proposed method and graphical visualization of the results for the most successful 200 queries can be seen on Figures 5.14 and 5.16 respectively. As seen in Figure 5.16, the black dots, corresponding to relevant images, are mostly on the left, showing that most of the words are matched with other instances of the same word. Figure 5.15 shows minimum, maximum and average mAP values for each unique word. As seen from the figure, words having much number of keypoints can have better mAP values, but difference between minimum and maximum mAP value increase for these words. This is because, writing styles can vary for long words, since they have more number of characters. In addition, these words are composed of two or more words and the proposed system can match two different words that have common words.

5.4.2 Retrieval Results for Ottoman Datasets

By evaluating the comparative results on IFN-ENIT-subset, we preferred to use Harris Affine detector rather than DoG detector and tf-idf represented feature



(a)

(b)

Figure 5.14: Two example queries for IFN-ENIT-subset. For each part, the image in the top left corner is the query word itself. Correct matches are indicated with a + sign on the top right corner of image. Note that in **(b)** the instances that are written with a thinner writing style are also retrieved, i.e. 15. image.

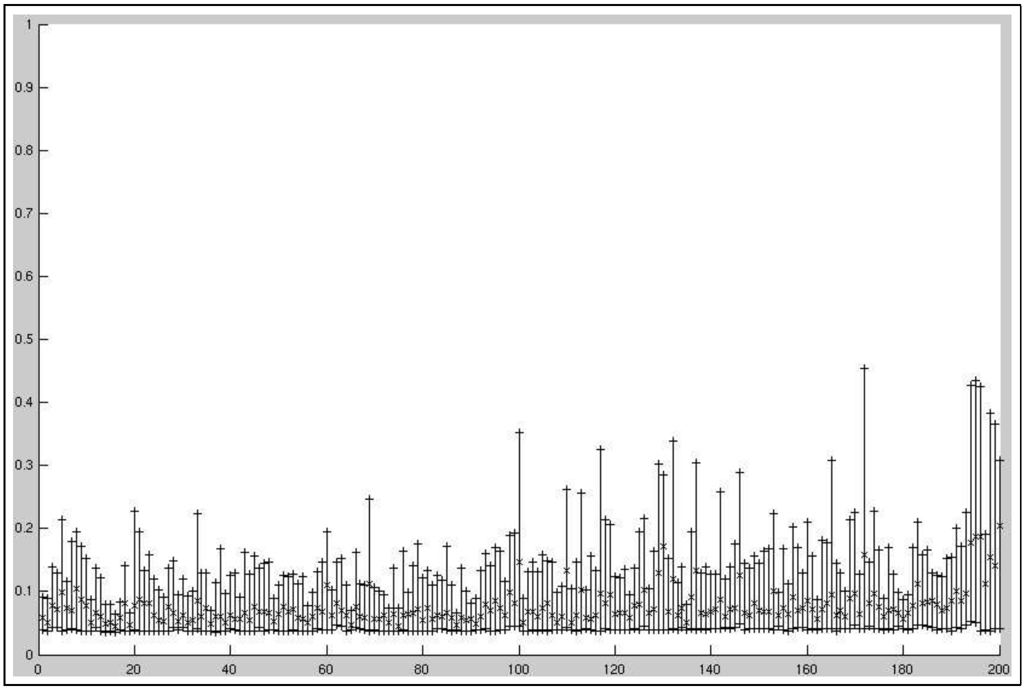


Figure 5.15: Minimum, maximum and average mAP results for 200 unique words of IFN-ENIT-subset. For each unique word minimum, average and maximum mAP value is displayed. The words are sorted along the x-axis in increasing order by their average number of keypoints and y axis represents mAP values. Note that for the words with much keypoints, maximum of the mAP values increase. But difference between minimum and maximum mAP value also increase. This is because these words are long words, so there can be more variances on the writing of instances of a word since it has more characters. In addition, long words generally compose of two or more words and the system can match two different words, that contain a common word.

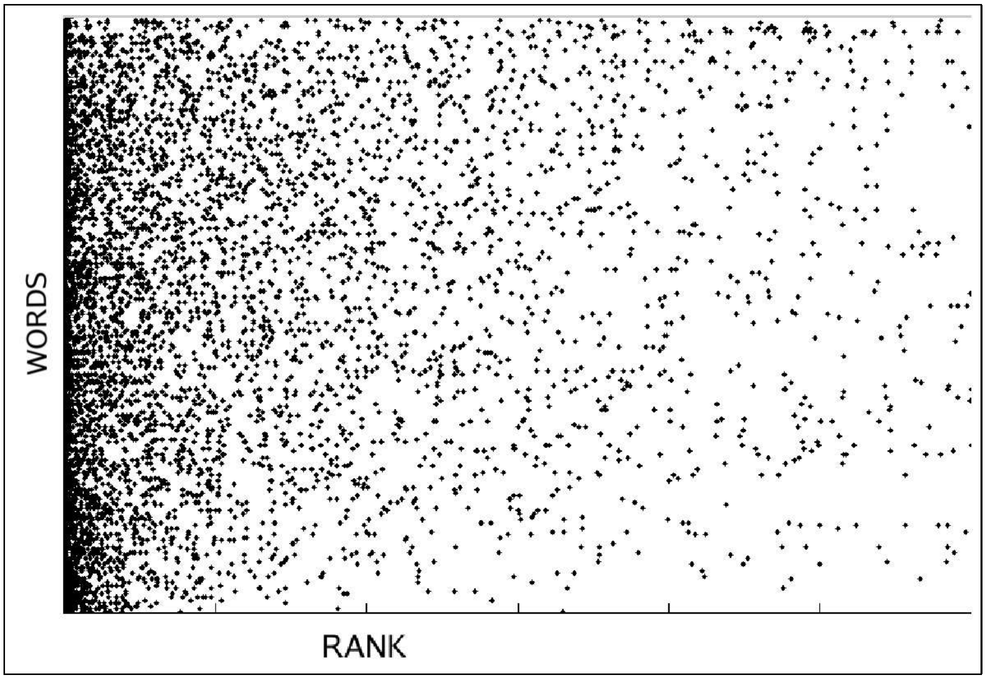
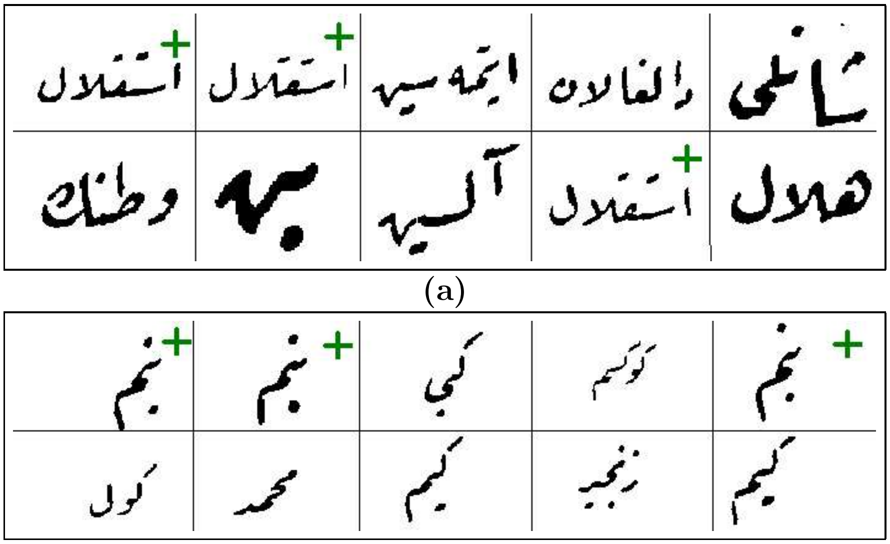


Figure 5.16: Results for the most successfull queries of each unique word in IFN-ENIT-subset. The words are sorted along the y-axis in increasing order by their mAP values and x axis represents the ranking order. A black dot indicates the relevant word-image to the query word. In ideal case, all black pixels are packed to the left.

vectors throughout the experiments on Ottoman dataset. The experiments are carried out on all of the three sets of Ottoman dataset. Since, small-printed and rika sets are manually annotated, quantitative results are obtained on these data sets in the form of mAP (mean Average Precision) values. In order to test the effect of mixed writing styles we also build another data set, which we refer as **combined**, by combining small-printed and rika data sets. Due to difficulty of annotating large-printed data set consisting of over 9500 words only qualitative results are given in this data set.

Figure 5.18 shows example query results for three words on large-printed data set and Figure 5.17 shows example query results for two words on rika data set. As figures show, other instances of the query words are correctly retrieved within the top matches.

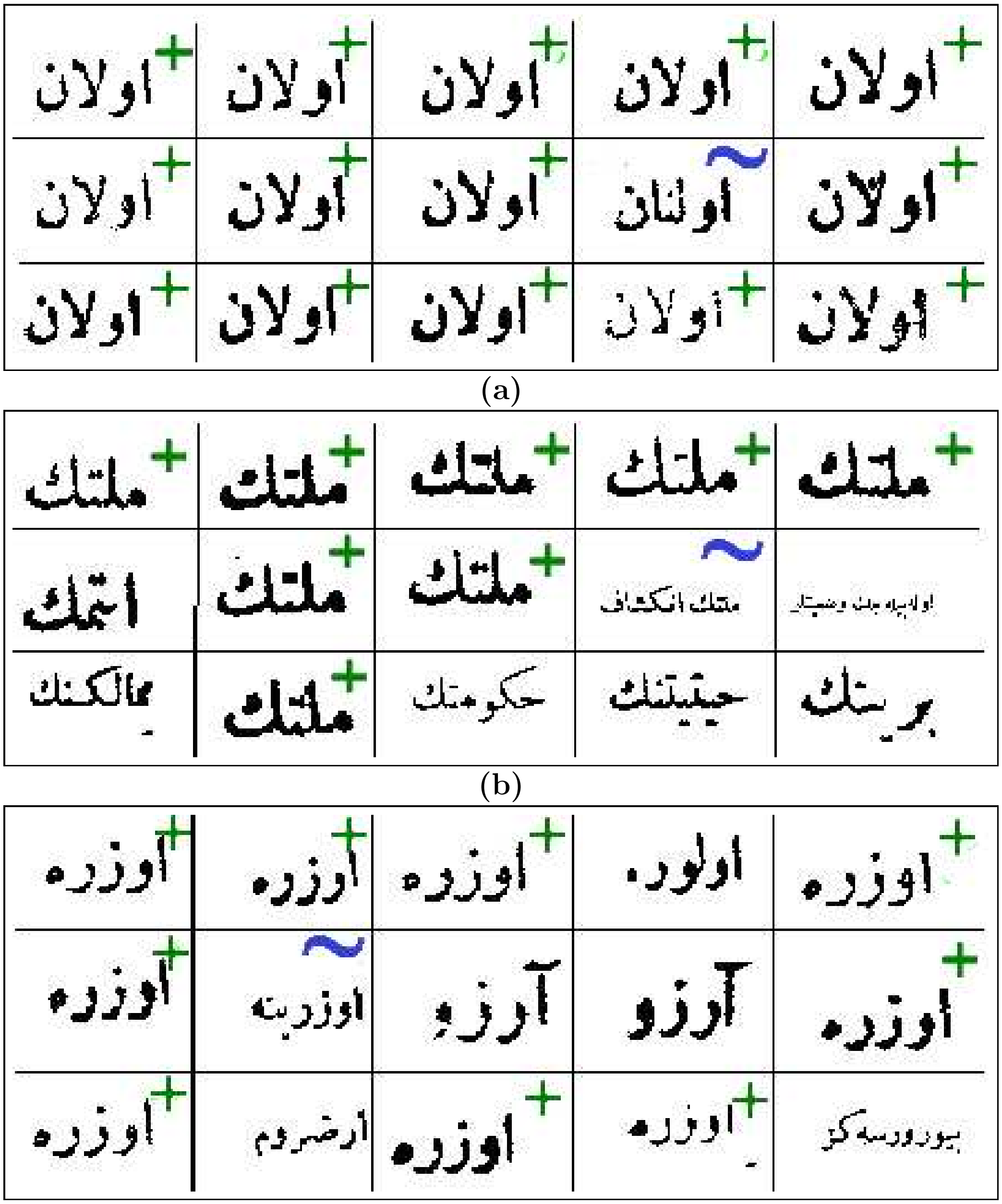


(a)

(b)

Figure 5.17: Example query results on rika data set. Exact matches are shown by $+$, and matched similar words are shown with $\sim$. **(a)** The first query word *istiklal* means *independence*, and **(b)** the second word *benim* means *my*. For the first query all three instances of the word are retrieved in the top ten. Note that retrieval results for short query words are worse as seen on **(b)**. Note also that retrieved images wholly resemble to query word and it is hard to distinguish irrelevant images from query image by eye.

Due to characteristics of Turkish language new words can be generated from a



(a)

(b)

(c)

Figure 5.18: Example query results for the first 15 matches on large-printed data sets. Exact matches are shown by green + sign, and matched similar words are shown with ∼. **(a)** The first query word *olan* means *existed*, **(b)** the second word *milletin* means *nation's* and **(c)** the thirds word *üzere* means *in this way.* Note that, in the queries, some semantically similar words are also retrieved.

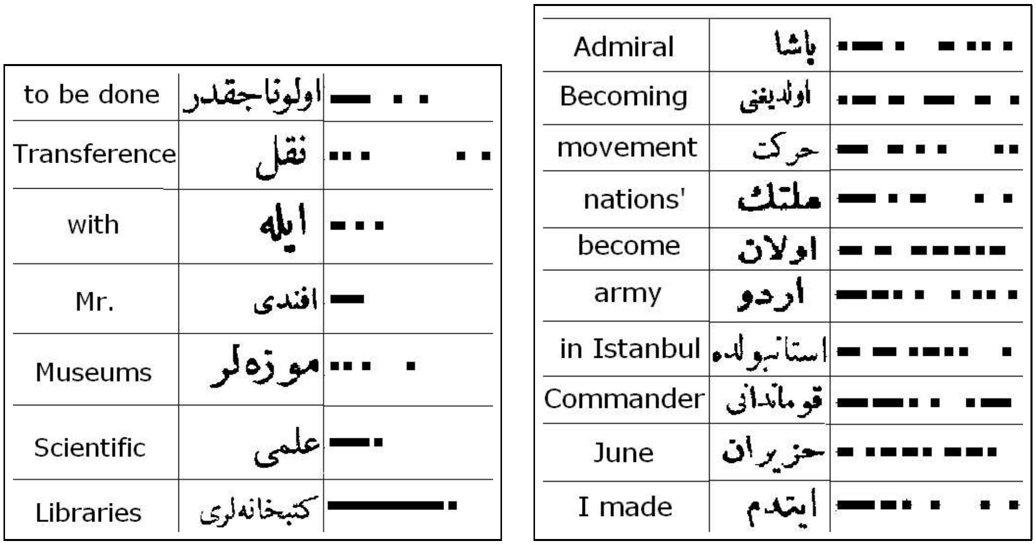


Figure 5.19: The ranked retrieval results for some selected words are shown on small-printed (left) and on large-printed (right) data sets. X axis shows the order of relevant documents retrieved in top 25 matches and Y axis represents the query words. In the ideal case, we expect all of the black dots on the left.

common stem using suffixes. Therefore, it is also important to match these words which are semantically similar. As can be seen from the figures, the proposed system is also able to capture the semantic similarities.

Figure 5.19 shows retrieval results for some selected words on small-printed and large-printed data sets. In this visualization, the words which are semantically similar are also considered as correct matches. As can be seen, the black dots, corresponding to relevant images, are mostly on the left, showing that most of the words are matched either with other instances of the same word or with semantically similar words with high accuracies.

Table 5.3 shows the mAP values on small-printed, rika and combined data sets. Clusters are obtained on each data set separately. Each word in a data set is used as a query and all the other words in that data set are ranked according to their similarity to the query word. The query word is always found as the first match, therefore the results when we use all words as query are higher (the number of words which appear only once is 465 for small-printed, 174 for rika, and

Table 5.3: mAP results for Ottoman datasets. Three cases are evaluated: **all words:** all of the words are used as query, **frequent** the ones which appear only once are skipped and the rest is used for query, **common:** only the words which are common in small-printed and rika are used for query.

	Small-Printed	Rika	Combined
All	.8892	.9080	.8748
Frequent	.7137	.7152	.6703
Common	.3383	.7206	.2592

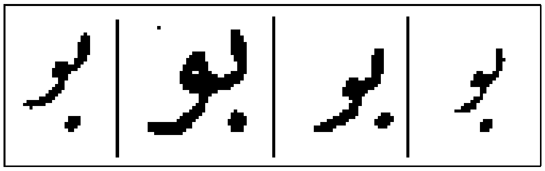


Figure 5.20: Example query results on combined data set, for one of the common words *bu* meaning *this*. The first one is the query word written in rika, and the others are retrieved words with rankings 12, 14 and 27. Note that second word which is in printed form is retrieved before the third and the fourth words which are in rika form, showing that the words written in different writing styles could also be matched. Note also the differences, especially in position of dots, for the words in rika, showing that slight differences are tolerated.

648 for combined). Note that, not the similar words but only the exact matches are considered for quantitative results.

There are only 10 words which are common in small-printed and rika, most of which are stop words, such as *one, each, what, which, like, all.* Since, most of these words are short words, the performance is low and when the data sets are mixed mAP values decrease further. However, as Figure 5.20 shows, we are still able to capture the similarities even for different writing styles. Therefore, we believe that the proposed system can be used to retrieve words written by different people in very different writing styles also for Ottoman documents.

Table 5.4: mAP values for different k values on GW-2 dataset. Right part is the results by using prunning. Note that 34 word images that does not include keypoints, are not queried in the system and mAP value is computed for the queries of remaining word images.

# cluster	mAP	performance	mAP	performance	recall
100	.4427	.2361	.6342	.4153	.7054
200	.4525	.2493	.6484	.4387	.7054
300	.4555	.2539	.6534	.4458	.7054
400	.4600	.2599	.6583	.4537	.7054
500	.4618	.2622	.6609	.4575	.7054
600	.4610	.2610	.6587	.4539	.7054
700	.4620	.2624	.6617	.4591	.7054

5.4.3 Retrieval Results for George Washington Dataset

Performance of our method for GW-1 is 0.3696 in mAP format. This set is in a worse quality than GW-2, having small word-images with few number of keypoints. So it has a smaller mAP value. Figure 5.21 shows some query results on GW-1.

We conducted experiments on GW-2 dataset with different vocabulary sizes and different techniques as if in IFN-ENIT-subset. As told previously, a method for guessing the optimum k-value on visterm generation is proposed for Ottoman dataset. George Washington dataset presents different characteristic with its alphabet and writing style. So we tried to find a suitable k value for GW-2 also. Table 5.4 shows retrieval results for different k-values. On these experiments we used tf-idf represented feature vectors and Harris-Affine detector and cosine distance for matching. By looking at the results we prefer to use a vocabulary size of 500 for GW-2.

Different detector and representation types are also experimented on GW-2 dataset. Results for different types of detectors, representations and distance measures are reported on Table 5.5 and Table 5.6. Similar to IFN-ENIT-subset, we also prefer to use Harris-Affine detector and tf-idf represented feature vectors with cosine distance on GW-2.

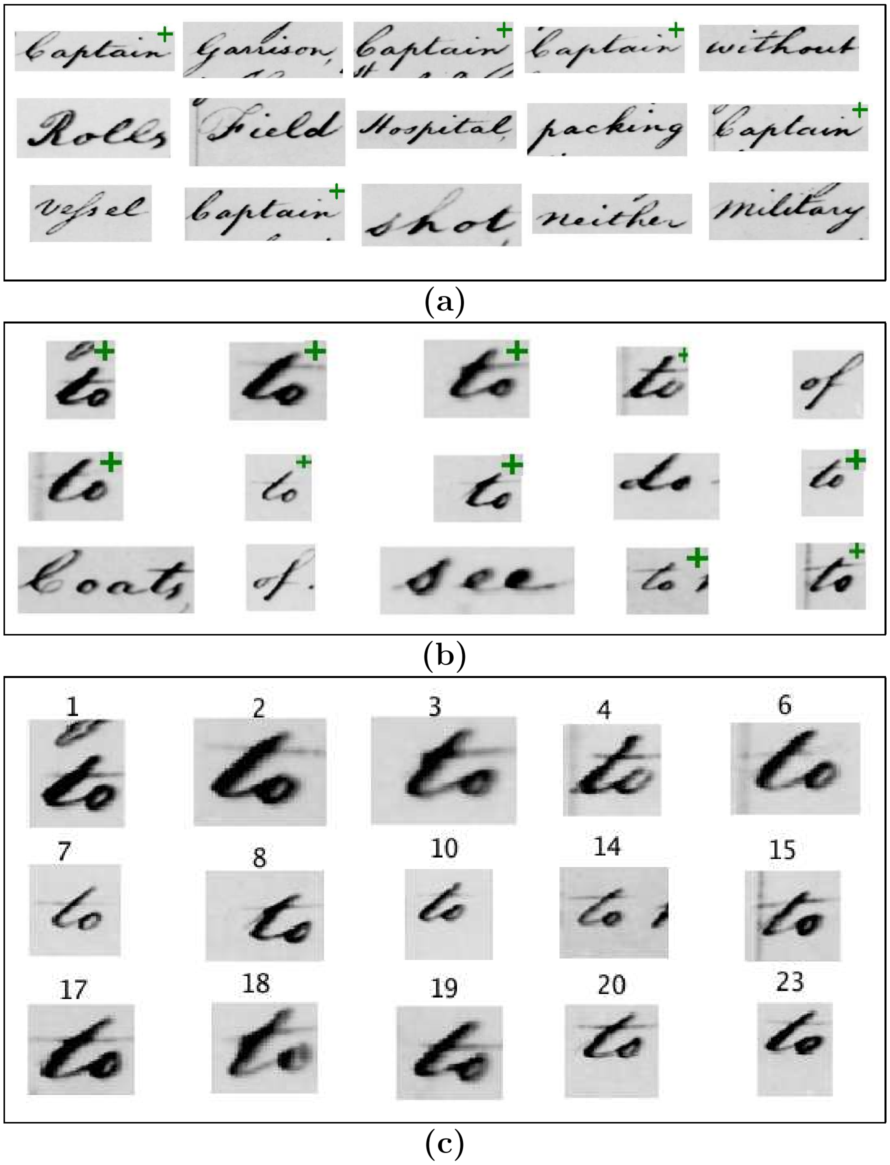


Figure 5.21: Some query results from GW-1, **(a)** first 15 results for query word *Captain*, **(b)** first 15 results for query word *to* and **(c)** first 15 correct matches for query word *to* with the ranking order on them.

Table 5.5: Results on GW-2 with $hist_{norm}$ and *tf-idf* represented feature vectors using Harris Affine and DoG detector. Note that, the best results are achieved for Harris Affine detector and tf-idf representation, hence we prefer to use them for further studies.

	Harris Affine		DoG
Distance Type	$hist_{norm}$	tf-idf	tf-idf
Cosine	.4610	.4618	.4694
KL-Divergence	.4202	.4208	.4240
Chi-Square	.4062	.4042	.4038
Euclidean	.4221	.4142	.4070
Weighted Inner Product	.3273	.3379	.4652
Binary Inner Product	.4265	.4265	.4120
XOR	.4222	.4222	.4313

Table 5.6: Results on GW-2 with Location Weighted Histograms and Soft Weighting (SoW) scheme. Note that after evaluating the results, we prefer to use tf-idf represented feature vectors.

Prime	$Base_2$Bins		$Base_2$Locs		SoW	
	Norm.	Not Norm.	Norm.	Not Norm.	1 neigh.	4 neigh.
.4040	.4077	.4075	.4143	.4141	.4757	.4750

5.5 Comparison with Other Methods

In order evaluate the performance of our system, we make some comparisons with other studies, which make use of our datasets also. These comparisons are performed for all of the datasets as explained in related sections in the following.

5.5.1 Comparisons on GW Dataset

GW-2 dataset is used by Rath et al. in [48]. After a pruning step, they run Dynamic Time Warping (DTW) algorithm with four features that are normalized vertical projection profile, normalized upper word profile and lower word profile and background-ink transition. So we also run our method for GW-2 dataset and compare the performances as seen in Table 5.7. First column shows recall and second column shows mAP value. Recall value is the fraction of relevant images retrieved and the reported value is the average recall among all queries. The third

Table 5.7: Comparative results on GW-2

Method	Recall	mAP	Performance
DTW [48]	.7110	.6534	.4098
Proposed method no pruning	1.0	.4618	.2622
Proposed method with pruning	.7054	.6609	.4575

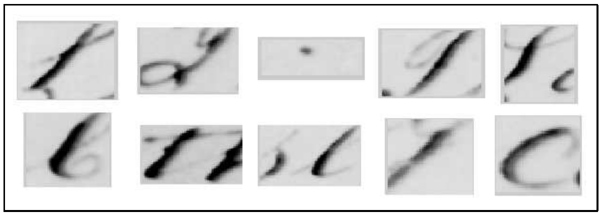


Figure 5.22: Images with zero keypoints that are not used as query for GW-2 dataset.

column shows the performance of system, which corresponds to mAP value when first matches, generally the query image itself, is not evaluated as relevant.

DTW method gives better results than the proposed method when no pruning step is used. We also run our algorithm with the same pruning step and acquire a closely similar mAP value to DTW method. The performance of our method with pruning step is very similar to the performance of DTW. Note that, slight difference on recall value comes from the images that are not queried in our system. These images does not have keypoints as they are miss-extracted nonword images that are shown in Figure 5.22.

Some query results of the proposed method with pruning step on GW-2 can be seen on Figure 5.24. Note that for long words, our system performs well, but for short words it is difficult to find matches since there are too few keypoints. Also, pruning step increases the performance of the proposed method, since it makes the system match the words having closer area and aspect ratio.

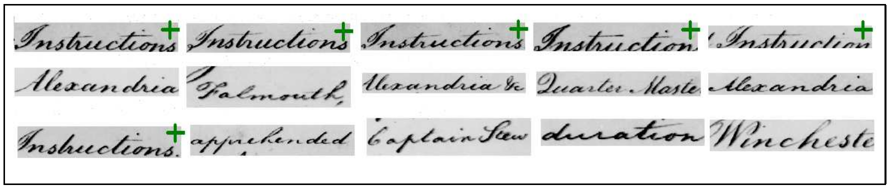
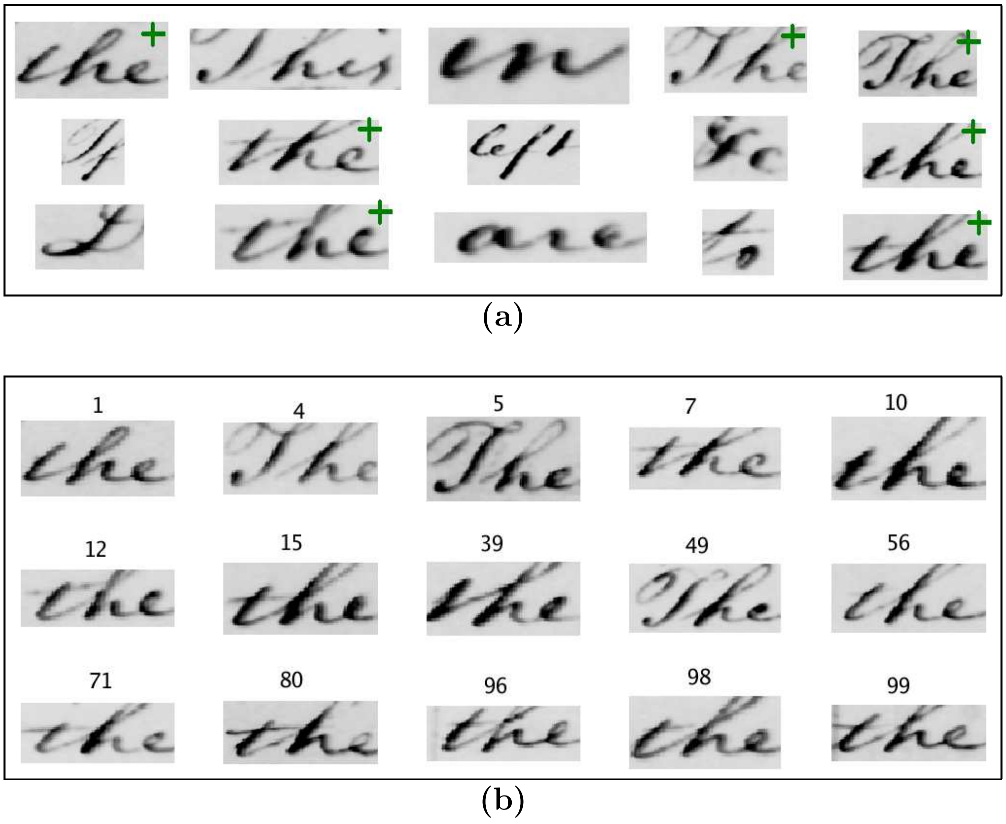


Figure 5.23: Query results for the word *Instructions* on GW-2.



(a)

(b)

Figure 5.24: Query results for the word *the* from GW-2. Part **(a)** shows the top 15 matches for the query word. Note that, some instances of the word with different writings are also retrieved. Part **(b)** shows top 15 relevant images with their ranking order on them.

Table 5.8: Results on selected queries of IFN-ENIT-subset. This sample subset includes 100 unique word images, which yield optimal results with the proposed methodology.

Method	mAP
DTW with VPP	.3282
Proposed method	.2019
String Matching	.1227
Lowe's SIFT Matching	.0068

5.5.2 Comparisons on IFN-ENIT and Ottoman Datasets

IFN-ENIT dataset is a more difficult dataset than GW datasets, since it has various writing styles and writers. From IFN-ENIT-subset, we selected a sample subset of 100 unique word images, which yield optimal results with the proposed methodology. We only compute the mAP values for these selected 100 queries. Hence we used this subset for comparison purposes, since running algorithms on the whole dataset takes much time.

We run Min Edit Distance algorithm, which is a kind of string matching algorithm, on the string represented version of feature vector. By this way, each image is represented with our visual vocabulary and the distance between the two images is taken as the number of operations to form one from another. As seen from the algorithm these operations are insertion, deletion and transposition, i.e. the words *ifnormaton* and *information* are 2 unit distant from each other since one substitution between letters *f* and *n* and insertion of letter *i* after *t* are required.

We also used Lowe's SIFT matching technique in order to find the similarities between the two images. Lowe's SIFT matching counts the number of similar keypoints between the two images. We used this value as the similarity between the two images.

Comparative results of experiments on the selected queries of IFN-ENIT-subset are presented in Table 5.8. DTW technique on vertical projection profile

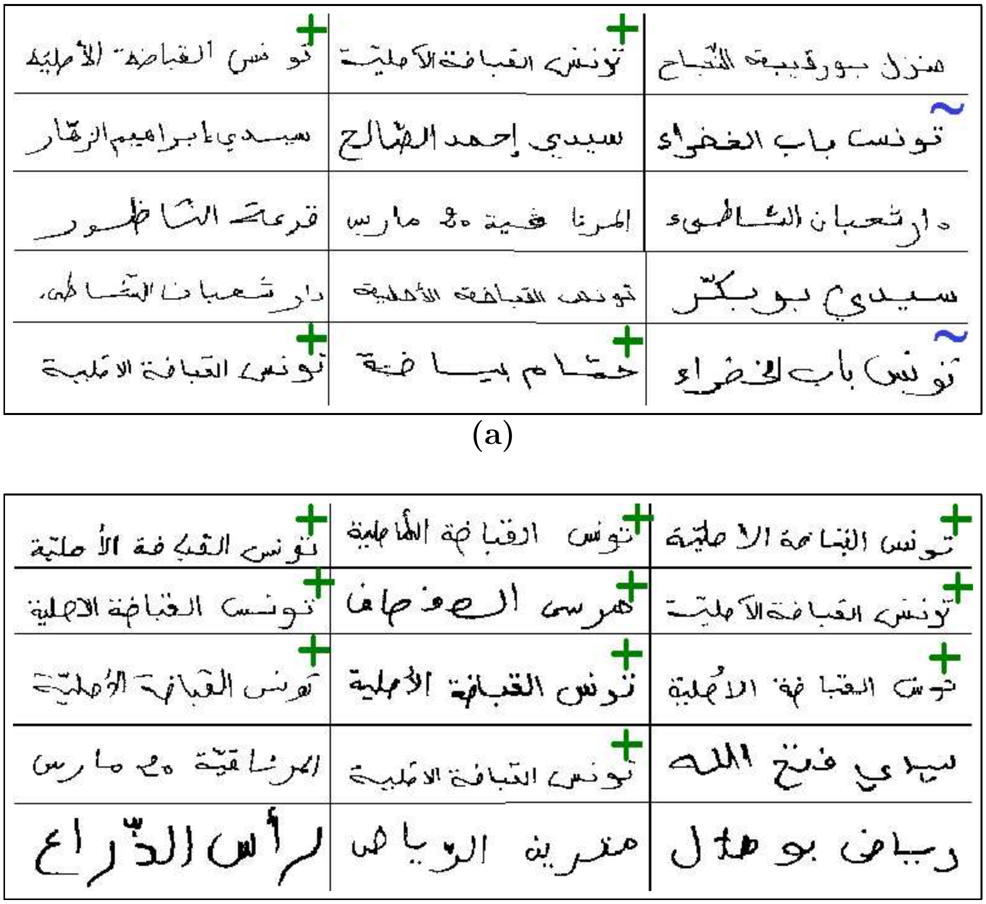
(*VPP*) of word images are carried out for this subset as well. Our method performs better than string matching and Lowe's SIFT matching. On the other hand, DTW with VPP outperforms our method. Tunisian city names are usually composed of two or more words and false matches of the proposed method are generally coming from the city names that have common words. Figure 5.25 shows query results for the same city name both with the proposed method and DTW technique. Notice that, recognition of the words in IFN-ENIT dataset is particularly difficult by eye.

DTW algorithm is a complicated algorithm with extensive running time. If the lengths of images are m and n, DTW runs in $O(m \times n)$ in order to find the distance between the images. On the other hand, our method finds the distance in $O(k)$ time where k is the length of feature vector. Thus, even if DTW gives a better performance on IFN-ENIT dataset, its running time is greater than the proposed method.

We also tried to use the same pruning step, that is applied to GW-2 dataset, for IFN-ENIT-subset, but since the set has various writers, instances of a word image can be in various lengths and aspect ratios. So usage of pruning step gives low recall value, resulting in misses in numerous correct matches. For the whole dataset, recall value is .3615 and mAP value is .2182, when pruning is used.

Similar to IFN-ENIT-subset, running DTW with VPP on small-printed dataset gives a better performance than our method. With DTW method we obtain mAP values 0.94 for all words and 0.86 for words appearing more than once. As reported in Table 5.3, the proposed method produces 0.89 mAP for all words and 0.71 mAP for frequent words on small-printed dataset. When compared with the table, the results of DTW method is better when mAP values are used for comparison. It is an expected result, since the DTW method is very successful in matching the exact instances of the words.

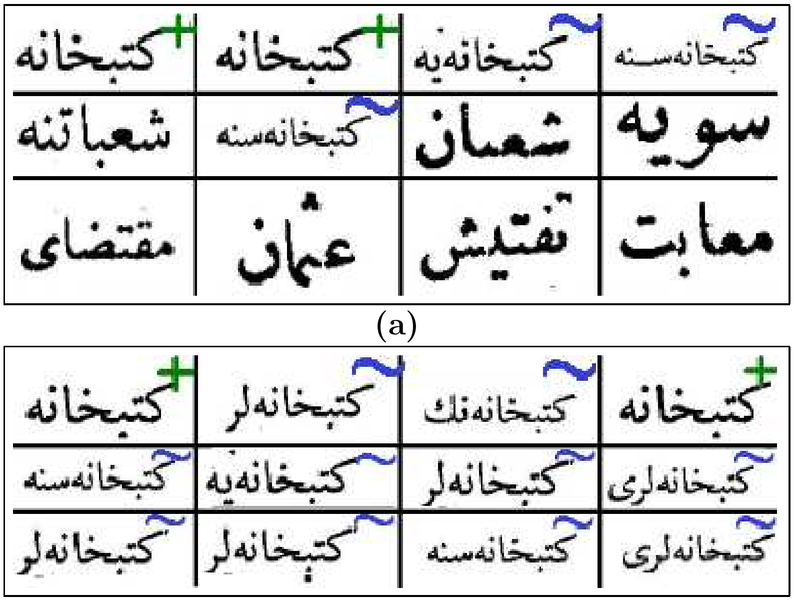
However, in Turkish new words are generated from a common stem using suffixes. For example, the words meaning *libraries, to library, to libraries, from library, at library* are all derived from a single stem meaning *library.* For a better retrieval of Ottoman documents these words should also be considered but DTW



(a)

(b)

Figure 5.25: **(a)** Query results with the proposed method, **(b)** query results with DTW method on VPP of word images for the same word. Exact matches are indicated with + signs and the city names, that have common words with the query name, are indicated with ∼. Query city name is *Tunus el-Kabâdat'el-Asliyya* and relevant ones indicated with ∼ are *Tunus Bâb'el-Khadrâ*.



(b)

Figure 5.26: A word *kütüphane*, meaning *library*, is queried both using DTW approach (a) and our system (b). Top 12 matches are shown. + signs indicate correct matches and ∼ signs indicate the semantically similar words, such as *libraries, to the library, public library* etc. Note that, all of the results are related to the query when the proposed system is used, while DTW is only able to capture 5 of the related words.

method is unable to capture these similarities as shown in Figure 5.26.

5.6 Indexing Results

The proposed representation is successful in retrieval and we also make use of it for word recognition. However, there are more refinements to do on usage of it for recognition as proposed representation is mostly succesful in word spotting which yields to indexing.

In [25] GW-1 dataset is used for recognition of word images with an HMM classifier. They used 19 pages for training and 1 page for testing each time through a *k-fold cross validation* where *k* is 20. Similarly, we train a k-nearest neighbor

Table 5.9: Recognition Results for GW-1

Method	Word Error Rate	Word Error Rate Excluding OOV Errors
HMM classifier [25]	.503	.414
K-NN classifier (k=9)	.785	.730

Table 5.10: Recognition Results for GW-1 with one-class classifiers

Type of classifier	Word Error Rate
Spectral Angle Mapper	.1804
Incremental SVDD	.3499
Nearest Neighborhood	.3616
K-NN with optimum k	.3696
kmeans with optimum k	.4276
Gaussian	.4870
Principal Component Analysis	.4960
Parzen Density	.4999
Mixture of Gaussian	.5000

classifier for GW-1 by using our feature vector. Recognition results are seen on Table 5.9. The results are given in the form of word error rate, that is average proportion of false matches. Since some errors are coming from test words that are not used in training, we also give the results excluding out of vocabulary (OOV) errors in the last column of table.

Examining the results, we prefer to use one-class classifiers and trained different types of classifiers for GW-1. GW-1 has 1464 unique words out of 4856 words and we trained 76 one-class classifiers for the words that exists more than or equal to 10. The results are seen on Table 5.10 and compared to the results of HMM classifier as used in the study of Rath *et al.* (Table 5.9) one-class classifiers that are trained with our feature vectors produce promising results for indexing.

As seen from the Table 5.10 Spectral Angle Mapper (SAM) classifier gives the best performance among all classifiers. SAM classifier tries to find a spectral angle for each class during training and assign each test sample to the class with minimum angle difference. Thus, direction of feature vector is more important than magnitude of it for SAM classifier.

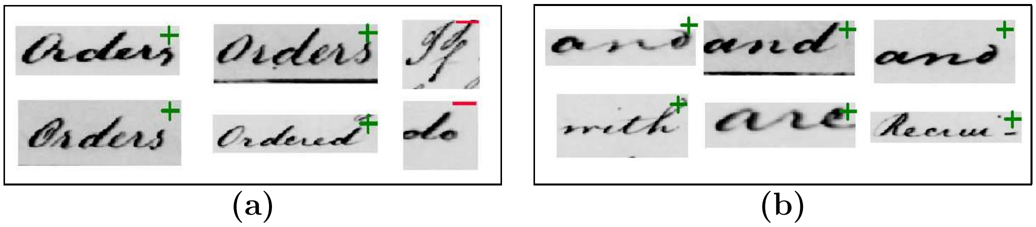


Figure 5.27: **(a)** Recognition result for the word *Orders*, **(b)** Recognition result for the word *and*. The words indicated with + are recognized as the target word and outliers are indicated with letter - on them. The system has difficulty in recognition of short words. Note that in part **(a)** the word *Ordered* is recognized as *Orders*, meaning the system is able to capture semantic similarities. Although this is not an exactly irrelevant case, we evaluate it as an false match during performance computation. In **(b)**, the word *are* is recognized as *and*, which visually resemble the other so much.

Table 5.11: Recognition Results for IFN-ENIT-subset

Type of classifier	Word Error Rate
One-class K-NN (k=5)	.2322
multi-class K-NN (k=5)	.8468

Figure 5.27 shows recognition results for two example words using SAMC. As if in retrieval, we have difficulty in the recognition of short words and some words, sharing a common root, can be recognized as same (i.e.*Orders* and *Ordered*). In addition, the system recognizes *at* as *to* or *so* as *as* showing that effective usage of location information can improve the recognition performace.

In order to see the performance difference between multiclass and one-class classifiers, we trained both of them for IFN-ENIT-subset. 10% of data is used for testing and the rest are used for training. Again we make use of k-fold cross validation by taking k as 10 and randomly selecting testing set in each run. Table 5.11 shows the recognition results in the form of Word Error Rate (WER) for one-class and multi class K-NN classifier. For one-class case average WER among all classifiers is reported. One-class classifiers give better results than the other as they cover and sketch feature space more succesfully.

IFN-ENIT-all is used in many studies for word recognition [46, 47]. We also

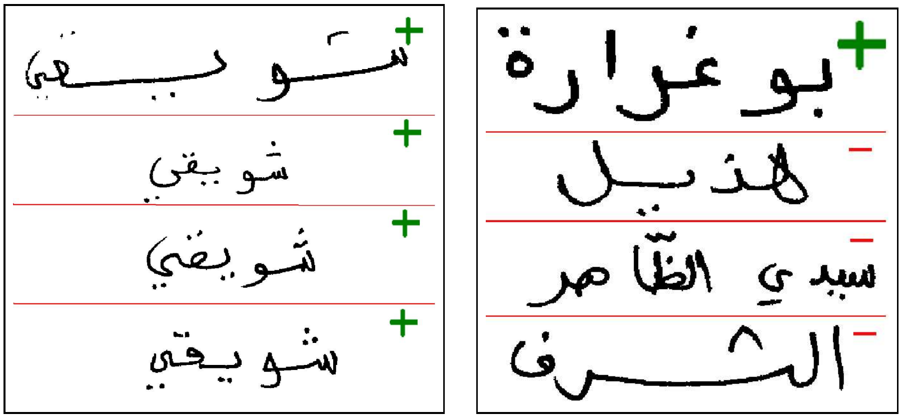
Table 5.12: Word recognition results on IFN-ENIT-all with one class SAMCs using our feature vector representation **(third column)** and HMM Classifier of [47] **(last column)**

Training Sets	Test Set	Recognition Rate	Recognition Rate [47]
a, b, c	d	.748	.985
b, c, d	a	.747	.984
c, d, a	b	.763	.986
d, a, b	c	.746	.982

used our proposed feature vector for recognition of the whole dataset. As stated in [45] the dataset is divided into 4 subsets (a, b, c and d) and one of them is used for testing while the others are the training samples. Similarly, we trained one-class SAM Classifiers for each city name and the results are reported in Table 5.12. For each test, 3 sets are used as training and the remaining one set is used for testing. Negative samples are chosen from related sets (i.e. training or test) randomly in the same amount with positive samples. The results are given in the form of recognition rate, which is the average proportion of correctly recognized words among all classifiers.

Pechwitz et al. [47] proposed a system performing recognition with about 98% accuracy for IFN-ENIT dataset. They used baseline information and some pixel features by generating a HMM model for each character shape. They used 160 different character shape models out of 28 letter Arabic alphabet. Hence, the system has got a heavy model generation and feature extraction step. Notice that our method is unsupervised and can be adapted to many languages easily. Comparative results are reported on Table 5.12 in the form of recognition rate, which is the proportion of words recognized correctly. The last column of the table shows the results of [47].

Figure 5.28 shows the results of SAM one class classifier for the word *shuyqi* from IFN-ENIT-all. As can be seen, the words that are recognized as target wrongly, resemble the target visually.



(a) (b)

Figure 5.28: Recognition results for the word *shuyqi* from IFN-ENIT dataset. **(a)** Positive test samples, **(b)** Negative test samples. The words indicated with + are recognized as the target word and outliers are indicated with letter - on them. The system can recognize the target word instances accurately. Note that, negative sample, which is recognized as target wrongly, has some elongated parts and dots resembling to the target word.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we proposed a novel framework for representing handwritten documents independent from the type of script. The main idea is approaching the problem like an unacquainted person and we have two main observations:

- A word is an image rather than a collection of characters
- Connection and curvature points and dots are important visual features for distinguishing two word images from each other

In text retrieval, documents are composed of words and terms give strong cues about the similarity of two documents. Term document relation of text retrieval systems can be adapted to our problem since a word image can be represented with some important visual words. Hence we make use of Bag-of-Features method, where each image is treated as document and visual terms are treated as words as if in text retrieval. Hence we can say that a word image is a collection of visual terms.

Based on these assumptions and observations, we detect salient regions of word images via Harris Affine detector and DoG detector. Detected keypoints are then described with SIFT features, where accurate results are achieved in many studies. In our case, visual terms, which we refer as visterms, are generated with vector quantization of keypoint descriptors. By this way, at the end of processes we know visterm information of each word image.

Different representations are created with this word image-visterm information. These representations are classical histogram of visterms, location weighted histogram of visterms and string representation of visual vocabulary. We also adapted term frequency-inverse document frequency approach of text retrieval to our problem and consequently among different representations we prefer to use tf-idf represented feature vector, since tf-idf increase the importance of rarely appearing terms and decrease importance of stop words.

The resulting representation is used for retrieval and indexing purposes with different types of distance measures and word spotting techniques. The experiments are carried out on documents of three different languages, Arabic, Latin and Ottoman and the results are reported in a comparative manner with other studies in this field. The results show that the proposed system is successful on retrieval and indexing independent from the language type.

Current character based techniques lack of covering all writing and scripting styles. On the other hand the proposed method is able to capture character similarities even if with different writing styles. Also, the system is capable of capturing semantic similarities, which is so important for indexing purpose. Besides, the proposed method does not include any supervising step and can be adapted to other languages easily as well.

One of the most recent studies on word matching is Dynamic Time Warping (DTW) technique. Performance of the proposed method is compared with DTW and the results show that the proposed method produces closely similar results to DTW method. While DTW is a time consuming technique, proposed method takes less time and does not include a supervising step. A comparison between HMM based character recognition techniques is also done and concluded that even if with an easy feature extraction process and without a heavy modeling, our system gives promising results, but there have to be done some improvements on the proposed representation in order to use it for recognition.

6.2 Future Work

The documents can be successfully represented with the proposed method. However, the proposed representation does not make use of location information of keypoints effectively. Although location information is not encoded, the system gives accurate results. In future studies, this representation can be refined with the use of location information and can give better results. Some other techniques can also be adopted to the problem, i.e. the word images can be represented as graphs with visterm information and word matching problem can be transformed to a graph matching problem.

Proposed system is independent from language. However, in case of adaption to a specific language, the representation can be refined with some additional features, that are special to that language. Moreover, if we know linguistic properties, the representation can yield to more accurate retrieval and recognition results as seen in many systems, which are based on specific languages.

For Ottoman script, the experiments are conducted on small scale datasets. However, the system can give better results for some other scripting styles of Ottoman script. For example, land register records of Ottoman empire are written with Siyaqat writing style, which is difficult to split into characters and this writing style offers a good example for applying our system. In our studies we could not use such kind of datasets, since we lack of a convenient dataset with a ground truth information. Thus, applying the proposed method to other calligraphy styles of Ottoman script remains open for further studies.

Our representation can also be used with other techniques as well. For example, as a future extension, proposed representation can be used in model generation of some recognizers like Hidden Markov Models or Artificial Neural Networks, which are generally used in documents analysis systems. This representation can also be used directly for character recognition rather than word recognition.

Bibliography

[1] I. S. I. Abuhaiba, S. A. Mahmoud, and R. J. Green. Recognition of Handwritten Cursive Arabic Characters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:664–672, 1994.

[2] T. Adamek, N. E. O'Connor, and A. F. Smeaton. Word Matching Using Single Closed Contours for Indexing Handwritten Historical Documents. *International Journal on Document Analysis and Recognition*, 9:153–165, 2007.

[3] B. Al-Badr and S. A. Mahmoud. Survey and Bibliography of Arabic Optical Text Tecognition. *Signal Processing*, 41:49–77, 1995.

[4] A. Amin. Off-line Arabic Character Recognition: - A Survey. *International Conference Document Analysis and Recognition*, 1997.

[5] A. Amin. Off-line Arabic Character Recognition: The State of the Art. *Pattern Recognition*, 31:517–530, 1998.

[6] N. Arica and F. T. Y. Vural. Bas: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognognition Letters*, 24:1627–1639, 2003.

[7] E. Ataer and P. Duygulu. Retrieval of Ottoman Documents. In *8th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2006.

[8] E. Ataer and P. Duygulu. Matching Ottoman Words: An Image Retrieval Approach to Historical Document Indexing. In *Proceedings of the International Conference on Image and Video Retrieval*, 2007.

[9] M. K. Atatürk. *Nutuk*. Türk Tayyare Cemiyeti, 1927.

[10] A. A. Atici and F. T. Yarman-Vural. A Heuristic Algorithm for Optical Character Recognition of Arabic Script. *Signal Processing*, 62(1):87–99, 1997.

[11] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:509–522, 2002.

[12] A. Benouareth, A. Ennaji, and M. Sellami. Hmms With Explicit State Duration Applied to Handwritten Arabic Word Recognition. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 897–900, 2006.

[13] J. Chan, C. Ziftci, and D. Forsyth. Searching Off-line Arabic Documents. In *In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[14] J. Edwards, Y. W. Teh, D. A. Forsyth, R. Bock, M. Maire, and G. Vesom. Making Latin Manuscripts Searchable Using gHMMs. In *Neural Information Processing Systems (NIPS)*, 2004.

[15] M. Eminoğlu. *Osmanli Vesikalarini Okumaya Giris*. Turkiye Diyanet Vakfi Yayinlari, 2003.

[16] A. Gillies, E. Erlandson, J. Trenkle, and S. Schlosser. Arabic Text Recognition System. In *Proceedings of the Symposium on Document Image Understanding Technology*, 1999.

[17] K. Gödel. Ueber Formal Unentscheidbare sätze der Principia Mathematica und Verwandter Systeme I (On Formally Undecidable Propositions of Principia Mathematica and Related Systems i). *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[18] N. Gök. Osmanlicayi Herkes Kolayca Ogrenebilir Mi, 2004.

[19] S. Impedovo, L. Ottaviano, and S. Occhinegro. Optical Character Recognition - A Survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5, 1991.

[20] S. F. J. L. Rothfeder and T. M. Rath. Using Corner Feature Correspondences to Rank Word Images by Similarity. In *Workshop on Document Image Analysis and Retrieval (DIAR)*, 2003.

[21] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards Optimal Bag-of-features for Object Categorization and Semantic Video Retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, 2007.

[22] A. Joly. New Local Descriptors Based on Dissociated Dipoles. In *Proceedings of the International Conference on Image and Video Retrieval*, 2007.

[23] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided Word Spotting in Historical Printed Documents Using Synthetic Data and User Feedback. *International Journal on Document Analysis and Recognition*, 9:167–177, 2007.

[24] Y. Kurt. *Osmanlica Dersleri 1.* Akcag Yayinlari, 2000.

[25] V. Lavrenko, T. M. Rath, and R. Manmatha. Holistic Word Recognition for Handwritten Historical Documents. In *DIAL '04: Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, pages 278–287, 2004.

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags ofFeatures: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[27] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[28] Library of Congress,
http://www.loc.gov/index.html.

[29] T. Lindeberg. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[30] L. M. Lorigo and V. Govindaraju. Offline Arabic Handwriting Recognition: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):712–724, 2006.

[31] D. G. Lowe. Distinctive Image Features From Scale-Invariant Keypoints. *International Journal on Computer Vision*, 60(2), 2004.

[32] R. Manmatha, C. Han, and E. M. Riseman. Word Spotting: A New Approach to Indexing Handwriting. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, 1995.

[33] R. Manmatha and T. Rath. Indexing of Handwritten Historical Documents - Recent Progress. In *Proc. of the Symposium on Document Image Understanding Technology (SDIUT)*, pages 77–85, 2003.

[34] J. Mantas. An Overview of Character Recognition Methodologies. *Pattern Recognition*, 19(6):425–430, 1986.

[35] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[36] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 27(10):1615–1630, 2005.

[37] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

[38] S. Mozaffari, K. Faez, and M. Ziaratban. Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 237–241, 2005.

[39] E. Nowak, F. Jurie, and B. Triggs. Sampling Strategies for Bag-of-features Image Classification. In *European Conference on Computer Vision*, 2006.

[40] A. Onat, F. Yildiz, and M. Gunduz. Ottoman Script Recognition Using Hidden Markov Model. In *IEEE Transactions on Engineering Computing Technology*, volume 14, 2006.

[41] Otap: Ottoman Text Archive Project,
http://courses.washington.edu/otap/.

[42] N. Otsu. A Threshold Selection Method From Gray Level Histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9, 1979.

[43] Ottoman Web Page,
http://www.osmanli700.gen.tr/.

[44] A. Ozturk, S. Gunes, and Y. Ozbay. Multifont Ottoman Character Recognition. In *The 7th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2000.

[45] M. Pechwitz, S. S. Maddouri, V. Megner, N. Ellouze, and H. Amiri. IFN/ENIT-database of handwritten Arabic words. In *7th Colloque International Francophone sur l'Ecrit et le Document , (CIFED)*, 2002.

[46] M. Pechwitz and V. Maergner. Hmm Based Approach for Handwritten Arabic Word Recognition Using the Ifn/enit- Database. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003.

[47] M. Pechwitz, V. Maergner, and H. E. Abed. Comparison of Two Different Feature Sets For Offline Recognition of Handwritten Arabic Words. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[48] T. M. Rath and R. Manmatha. Word Image Matching Using Dynamic Time Warping. In *CVPR (2)*, pages 521–527, 2003.

[49] J. L. Rothfeder, R. Manmatha, and T. M. Rath. Aligning Transcripts to Automatically Segmented Handwritten Manuscripts. In *Document Analysis Systems*, pages 84–95, 2006.

[50] E. Saykol, A. K. Sinop, U. Gudukbay, O. Ulusoy, and A. E. Cetin. Content-based Retrieval of Historical Ottoman Documents Stored as Textual Images. *IEEE Transactions on Image Processing*, 13, 2004.

[51] L. Schomaker and E. Segers. Finding Features Used in the Human Reading of Cursive Handwriting. *IJDAR*, 2:13–18, 1999.

[52] S. A. Shahab, W. G. Al-Khatib, and S. A. Mahmoud. Computer Aided Indexing of Historical Manuscripts. In *Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation*, 2006.

[53] A. Sisman and F. Yarman-Vural. Ottoman Transcription System. In *ISCIS-IX*, 1996.

[54] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003.

[55] S. N. Srihari, H. Srinivasan, P. Babu, and C. Bhole. Spotting Words in Handwritten Arabic Documents. In *Proc. Document Recognition and Retrieval XIII (SPIE)*, 2006.

[56] Suen, C. Y., Berthod, Marc, Mori, and Shunji. Automatic Recognition of Handprinted Characters - The State of the Art. In *Proceedings of the IEEE*, volume 68, pages 469–487, 1980.

[57] D. M. J. Tax. *One-class Classification; Concept-learning in the Absence of Counter-examples*. PhD thesis, Delft University of Technology, 2001.

[58] Turkish Republic Government Archive Web Page,
http://www.devletarsivleri.gov.tr/.

[59] M. Ülker. *Baslangictan Günümüze Türk Hat Sanati*. Turkiye Is Bankasi Kültür Yayinlari, 1987.

[60] J. R. Ullmann. Advance In Character Recognition. In *Application of Pattern Recognition*, 1982.

[61] F. Yarman-Vural and A. Atici. A Segmentation and Feature Extraction Algorithm for Ottoman Cursive Script. In *The Third Turkish Symposium on Artificial Intelligence and Neural Networks*, June 1994.