A COMPARISON OF METHODS FOR TRADEMARK RETRIEVAL IN A LARGE TRADEMARK DATASET

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

WUSIMAN TUERXUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY 2015

Approval of the thesis:

A COMPARISON OF METHODS FOR TRADEMARK RETRIEVAL IN A LARGE TRADEMARK DATASET

submitted by **WUSIMAN TUERXUN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı ________________
Head of Department, **Computer Engineering**

Assist. Prof. Dr. Sinan Kalkan ________________
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Fatoş Yarman-Vural
Computer Engineering Department, METU ________________

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU ________________

Assoc. Prof. Dr. Alptekin Temizel
Graduate School of Informatics, METU ________________

Assist. Prof. Dr. Yusuf Sahillioğlu
Computer Engineering Department, METU ________________

Assist. Prof. Dr. Erkut Erdem
Computer Engineering Department, Hacettepe University ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: WUSIMAN TUERXUN

Signature :

ABSTRACT

A COMPARISON OF METHODS FOR TRADEMARK RETRIEVAL IN A LARGE TRADEMARK DATASET

Tuerxun, Wusiman
M.S., Department of Computer Engineering
Supervisor : Assist. Prof. Dr. Sinan Kalkan

February 2015, 70 pages

By the end of the first decades of the 21^{th} century, the applications for trademarks worldwide have approached an astounding number of 5 million. In Turkey alone, this number has reached the 1 million mark, and is expected by all means to keep increasing. The accelerating competition of trademark influence and uniqueness has intensified trademark piracies and infringements, and therefore resulted in a substantial burden for the patent offices, not to mention direct economic losses. To overcome this ever-increasing problem of trademark registration without compromising from service quality, and to minimize unnecessary disputes of legal ownership, organizations like patent offices gradually turn to automatized registration, employing Trademark Retrieval Systems (TRS) equipped with image processing and computer vision tools. The last two decades have seen the successful implementation of well-known content-based image processing (CBIR) techniques in several TRS systems. However, these results are falling behind as the rapid increase in trademark registration escalates the trademark retrieval problem to the next level. Developing next-generation TRS systems with well-examined and analyzed new image retrieval and object detection techniques is necessary. Yet, the lack of public large-scale trademark datasets has obstructed the progress of this research field. In this thesis, to fill this gap, we offer a large scale and challenging dataset with $\sim$ 1 million trademarks as a benchmark. Then, as an initial attempt to trademark retrieval research on large scale datasets, we implement and analyze a variety of global image descriptors (*e.g.*, color, shape, texture), as well as local image descriptors (*e.g.*, SIFT, SURF, HOG, ORG), on this dataset (called the METU trademark dataset).

Keywords: trademark recognition, trademark retrieval, large-scale trademark dataset

ÖZ

BÜYÜK ÖLÇEKLİ MARKA VERİ KÜMELERİNDE MARKA TARAMA YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Tuerxun, Wusiman
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Sinan Kalkan

Şubat 2015 , 70 sayfa

21. yüzyılın ilk on yılları itibariyle, dünya çapında yapılmış olan marka tescil başvuruları sayısı etkileyici bir rakam olan 5 milyona ulaşmıştır. Sadece Türkiye'de, bu sayı 1 milyon barajını geçmiştir ve giderek artmaya devam edeceği de kesindir. Markaların reklam etkisi ve özgünlüğü konusunda gittikçe hızlanan rekabet ortamı, marka korsanlığının ve ihlallerinin artmasına neden olmuş, dolayısıyla da hem ekonomik zarara, hem de patent ofislerinde ciddi bir iş yüküne sebebiyet vermiştir. Gittikçe artan marka tescil başvurusularındaki bu problemi hizmet kalitesinden ödün vermeden çözebilmek için, patent ofisleri gibi kurumlar otomatik Marka Tarama Sistemlerine (*ing.* Trademark Retrieval Systems, TRS) geçiş yapmakta, bu sayede bilgisayarlı görüntü işleme tekniklerinden faydalanmaya başlamaktadır. Son yirmi yılda, yaygın içerik tabanlı görüntü işleme (*ing.* content-based image processing, CBIR) tekniklerini kullanan Marka Tarama Sistemleri oluşturulmuştur. Bununla birlikte, marka tescil başvurularındaki hızlı artış problemi yeni bir boyuta taşımakta ve bu sistemler gün geçtikçe yetersiz kalmaktadır. Sistematik olarak analiz edilmiş yeni görüntü tarama ve nesne saptama yöntemleriyle güçlendirilmiş yeni kuşak marka tarama sistemlerinin oluşturulması bu açıdan aciliyet arz etmektedir. Ancak, genele açık büyük ölçekli bir marka veritabanının bulunmayışı, bu alandaki ilerlemeyi yavaşlatmıştır. Bu tezde, literatürdeki bu boşluğu doldurmak amacıyla, 1 milyon kadar markadan oluşan büyük ölçekli bir marka veritabanı sunulmaktadır. Ayrıca, büyük ölçekli veritabanlarında marka tarama konusunda bir ilk teşebbüs olarak, çeşitli global görüntü

tanımlayıcıları (*örn.*, renk, şekil, desen) ve lokal görüntü tanımlayıcıları (*örn.*, SIFT, SURF, HOG, ORG), ODTÜ marka verikümesi adını verdiğimiz bu veri kümesi üzerinde çalıştırılarak analiz edilmektedir.

Anahtar Kelimeler: marka tanıma, marka tarama, büyük ölçekli marka veri kümesi

To whom have enlightened, loved, and believed me and my precious youthhood

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Assist. Prof. Dr. Sinan Kalkan for his patience, enthusiasm and continuous support. Under his guidance, I successfully finished my thesis.

Besides, I would like to acknowledge the rest of my thesis committee: Prof. Dr. Fatoş Yarman-Vural, Assoc. Prof. Dr. Alptekin Temizel, Assist. Prof. Dr. Yusuf Sahillioğlu and Assist. Dr. Prof. Erkut Erdem for their encouragement, insightful comments, and instructive guidances.

In addition, my completion of this thesis could not have been accomplished without the academic and moral support of my labmates, Hande Çelikkanat, Fatih Gökçe, Güner Orhan and Mehmet Akif AKKUŞ. When I joined the KOVAN Research Lab, your warm reception help me to adapt myself to the new working environment quickly. And I was being troublesome and curios guy among us, but you were always patient to my curiosity and never hesitate to share your knowledge and resources with me. Although I was living alone in Ankara, the company of you and your families and friends made me feel at home. During these two years, I have got through many tough times with your help. Especially, Hande, you helped me a lot during the most difficult times before and after the dissertation. Thank you very much.

Last but not the least, I would like to express my deepest thanks and appreciations to my family. They gave me constant encouragement, selfless support, and kindly solicitude. Here, I would like to share all my achievement with them.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

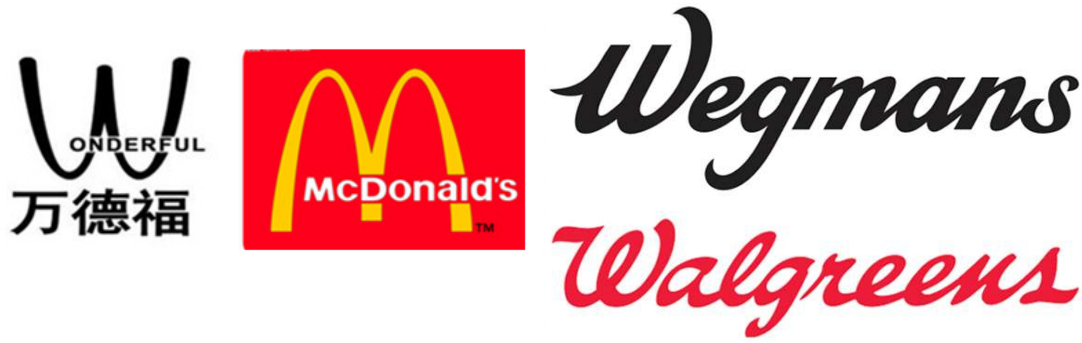
LIST OF ABBREVIATIONS

BRIEF	Binary robust independent elementary features
ORB	Oriented FAST and rotated BRIEF
SIFT	Scale-invariant feature transform
CBIR	Content based image retrieval
TRS	Trademark retrieval system
BoVW	Bag of Visual Words
TF-IDF	Term frequency-inverse document frequency
HOG	Histogram of Oriented Gradients
SURF	Speeded Up Robust Features
TPI	Turkish Patent Institute
WIPO	World Intellectual Property Organization
TR	Trademark Retrieval
LBP	Local Binary Pattern
IRP	Inverse Rank Position
BC	Borda Count
LO	Leave Out

CHAPTER 1

INTRODUCTION

A trademark is a recognizable symbol or associated text that identifies products or services of an individual, a business organization or a legal entity from those of others. Registered trademarks are viewed as a form of legitimate property and protected from brand piracy and trademark infringement. To protect and legalize their trademarks, owners have to register their trademarks in patent offices in many countries. More than 100 million companies are known to exist in local and global markets[1], and many of them own at least one registered trademark. According to Word Intellectual Property Organization, 2014, [60] 3 million trademark registrations exist worldwide and trademark applications keep increasing at a rate of 6-8% in recent two years. Economically fast growing developing countries such as Turkey and China have tremendous contribution for the increase in new registrations and applications. In Turkey, more than 1 million registered trademarks will be available by 2015 according to Turkish Patent Institute (TPI) yearly statistics [34]. As a consequence, the rapid increase of trademark registrations in local and global trademark patent office has challenged the Trademark Retrieval (TR) systems' "no duplication" property: any registered trademark in registration system should be different from other trademarks in both semantic and visual aspect. Increase in trademark infringements or likelihood of confusions could cause economical and social damages, and therefore should be resolved.



(a) McDonald vs. Wonderful (Adapted from [74]) (b) Walgreens vs. Wegmans (Adapted from [49])

Figure 1.1: Sample cases of trademark infringement.

In most developed countries, organizations like patent offices take the responsibility of protecting trademarks from encroachment. To avoid various infringements, they exclude registration

[1] See `http://www.econstats.com/wdi/wdiv_494.html` for related statistics.

of near-duplicated or intentionally imitated trademarks by manually checking trademarks in the database or by using TR systems. Massive amounts of registration have overwhelmed both manual and automatic operations and reduced service quality of patent offices, which leaves an open space for trademark infringements. What is worse, two mistakenly registered similar trademarks will increase the complexity of handling legal disputation between owners. To ease the burdens of patent offices, a robust TR system with intelligent image analyzing techniques is imperative.

The new generation TR systems should be equipped with excellent similarity detection tools able to work on large scale trademark datasets. In spite of recent progress on object detection and retrieval, evaluating trademark similarity from images is challenging because the source of similarity varies from low-level information like curvatures, corners (and their combinations) to high-level information requiring a semantic interpretation of the content of the trademarks; Besides, as mentioned above, the amount of registered trademarks is tremendous and new registrations are continuing due to rapid economic development. In the last two decades, adapting content-based image retrieval (CBIR) techniques to trademark similarity retrieval problem has improved the field to some degree. Such TR systems offered a valuable study platform to related studies of CBIR for several reasons, including: 1) Trademark retrieval techniques bear great commercial value. 2) Compared to other image datasets, trademark image datasets are well-organized and resourceful. 3) Trademark similarity problem includes various baffling image processing and perception problems.

1.1 Contributions

As discussed above, trademark retrieval is an interesting, challenging and important problem for both economic and academic value. However, the TR literature has failed to provide a challenging large-scale trademark dataset on which methods can be compared and tested throughly. For this goal, we have made two main contributions in this thesis:

- Firstly, we provide a large scale dataset of 930,328 trademarks as a benchmark for trademark retrieval studies [2] [81]. Although a number of trademark datasets are available in the literature, as will be reviewed later, most of them are of small scale and they lack variety. Our dataset not only has a large number of images but also has well-designed query sets and trademarks with various types. It includes 930,328 unique logo images, and 320 of them are composed of 32 sets of similar trademarks. Table 1.1 shows several datasets in literatures.
- Secondly, in spite of several studies on trademark retrieval, a detailed analysis and comparison between well-established object detection and retrieval algorithms is still missing. To close the gap with our best effort, in this thesis, we implement well-known local and global descriptors based on color, texture and shape; namely, color histogram, shape context, LBP, SIFT, SURF, ORB, etc. On the dataset, we provide a comprehensive analysis of the methods from different performance aspects.

A part of the work explained in this thesis is submitted to:

[2] This dataset is accessible for research purposes at `http://kovan.ceng.metu.edu.tr/LogoDataset/`

Table 1.1: Comparison of publicly available trademark datasets.

Dataset	**Dataset Type**	**Number of logos (and images)**	**Types of images**	**Year**
University of Maryland (UMD) [55]	Logo-to-logo	106 (-)	Bi-color	2001
BelgaLogos [38]	Logo-to-image	26 (10,000)	RGB	2009
Flickr Logos [66]	Logo-to-image	32 (8,240)	RGB	2011
MICC Logos [72]	Logo-to-image	13 (720)	RGB	2013
MPEG7 CE Shape-2 Part- B	logo-to-logo	3621	Bi -color	unknown
METU Trademark Dataset	**Logo-to-logo**	**409,834 (930,372)**	**RGB**	**2014**

Osman Tursun, Sinan Kalkan, A Challenging Big Dataset for Benchmarking Trademark Retrieval, 14^{th} *IAPR Conference on Machine Vision and Applications*, 2015.

1.2 Organization

The rest of the thesis is organized as follows:

Chapter two is mainly about background and related literature. In this chapter, we introduce trademark and its classification, registration, similarity and infringements. Besides these we also introduce design and structure of TR systems. At the end of the chapter, we discuss related studies in detail.

In chapter three, we present a brief introduction of various descriptors for TR. Certain methods among them are precisely introduced because of their outstanding performance comparing to other similar methods in most cases. Moreover, at the end of the chapter, we discuss several retrieval strategies of the trademark system.

In chapter four, we present our large scale trademark dataset and a comparison of other related datasets.

In chapter five, the setup and configurations of our experiments is given. Addition to that we analyze and compare results based on statistical evaluation and visual outputs of our experiments.

Chapter six is conclusions and future work.

CHAPTER 2

TRADEMARKS AND THEIR RETRIEVAL

In this chapter, we will review the background and relevant studies on trademarks and Trademark Retrieval Systems (TRS) in detail.

2.1 Trademarks and their usage

In this section, we will introduce the origin, the definition, the types and the similarity problem of trademarks. In addition, we will provide trademark statistics and illustrate the trademark similarity problem with several examples.

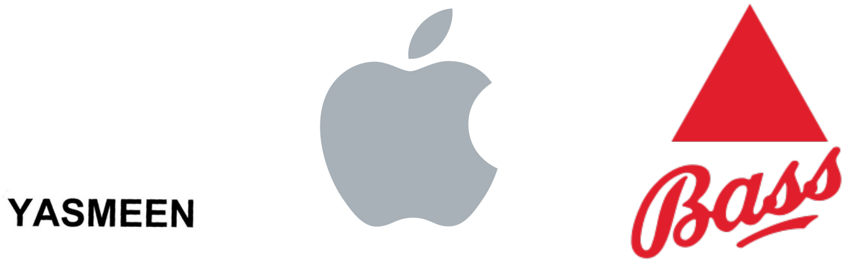
2.1.1 Trademark: definition and types

A trademark is a recognizable symbol of an organization or individual, which exclusively identifies its products, services or ideology from those of others. From a macro perspective, any modality like picture, sound, movement, which is capable of distinguishing goods, services, will be included in the trademark category. In this thesis, however, we look at trademarks at a micro angle, where we refer to figures or texts by the term *trademark*. The *trademark* is also referred to as *logo*, *service mark*, *brand*, *mark* [42] in related literature. However, the term *trademark* is adopted in this thesis after [36, 42, 86].

Trademarks are figures which could be composed of characters, shapes, textures or a combination of them. Based on the composition, trademarks can be categorized into text-only mark, a figure-only mark or a figure and text mark (see Figure 2.1 for samples). Text-only mark consists purely of text words or phrases. On the other hand, for a figure-only mark, the trademark only contains symbols, shapes, icons or images. When a trademark design includes both text and any non-textual information, it can be regarded as a figure and text mark [36, 86].

2.1.2 Trademark registration

To designate the ownership of goods or services, trademarks are used since from the fourth millennium BC [42]. Near the fall of the Roman empire, trademarks were used for arms and products, which represented the reputation of the owner. The first trademark legislation was registered in England in 1266, and the early modern laws on trademark registration were



(a) Text only mark (b) Figure-only mark (c) Figure and text mark

Figure 2.1: Examples of different trademark types.

established in the late 19^{th} century in Europe and early 20^{th} century in North America. The Bass Brewery's logo (Figure 2.1c) has been the first image to be registered as a trademark in 1875.

The trademark laws consider a trademark to be a form of property. Trademark offices (or “trademarks registry") are then in charge of protecting trademark propriety rights. In Turkey TPI[1] (Turkish Patent Institute) provides trademark protection service, while, for the global market, international registration systems like Madrid System[2] are used.

2.1.3 Trademark similarity

A trademark must go through a comprehensive checking process to eliminate the near-duplicate or likelihood of confusion compared to previously registered trademarks. The decision is established on a similarity degree, which, however, is not trivial since similarity is an ambiguous concept. For this reason, Indian, UK and European laws all consider the case of “likelihood of consumer confusion" for analyzing trademark infringements [8, 42].

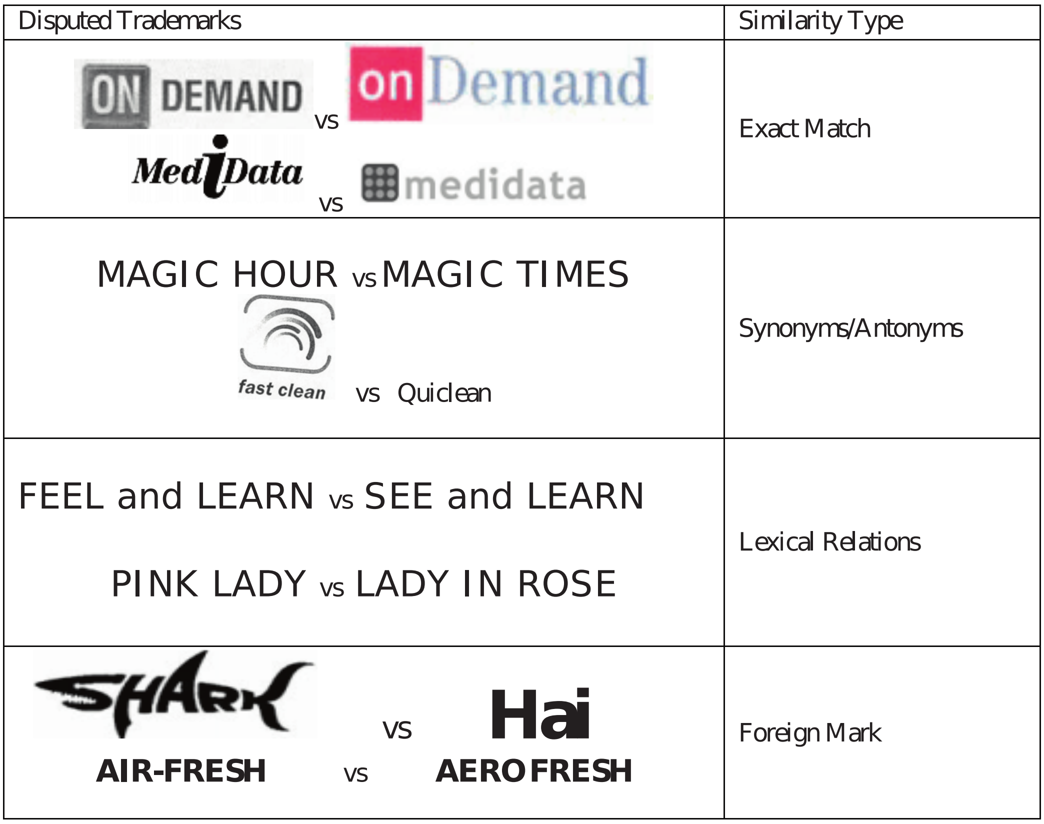
Similarity between trademarks can be either visual (at the level of visual information - see Figure 2.3 for samples) or conceptual (the semantic content that the trademark suggests) [8, 42]. Although visual similarity can be determined using visual information such as shape, topology, color etc. and for semantic similarity, for the time being, Vienna annotation labels are used in the literature.

Anuar et al. [8] suggested that conceptual similarity can be divided into four groups: *exact match*, *synonyms/antonyms*, *lexical conceptual relation* and *cross-lingual synonyms*, which are exemplified in Figure 2.2. In addition to these, trademarks can be similar at the level of phonemes. In Table 2.1, we display several phonetic similarities, such as "VERSACE"-vs-"NURSACE", "VEGETA"-vs-"ARGETA" and "SONY"-vs-"SQNY".

Although conceptual similarity is one of the main challenges of trademark retrieval, in this work, we only focus on visual similarity of trademarks.

[1] `http://www.tpe.gov.tr/TurkPatentEnstitusu/`

[2] `http://www.wipo.int/madrid/en/`



Disputed Trademarks	Similarity Type
ON DEMAND vs on Demand MediData vs medidata	Exact Match
MAGIC HOUR vs MAGIC TIMES fast clean vs Quiclean	Synonyms/Antonyms
FEEL and LEARN vs SEE and LEARN PINK LADY vs LADY IN ROSE	Lexical Relations
SHARK vs Hai AIR-FRESH vs AEROFRESH	Foreign Mark

Figure 2.2: Four types of conceptual similarity (Adapted from from [8]).

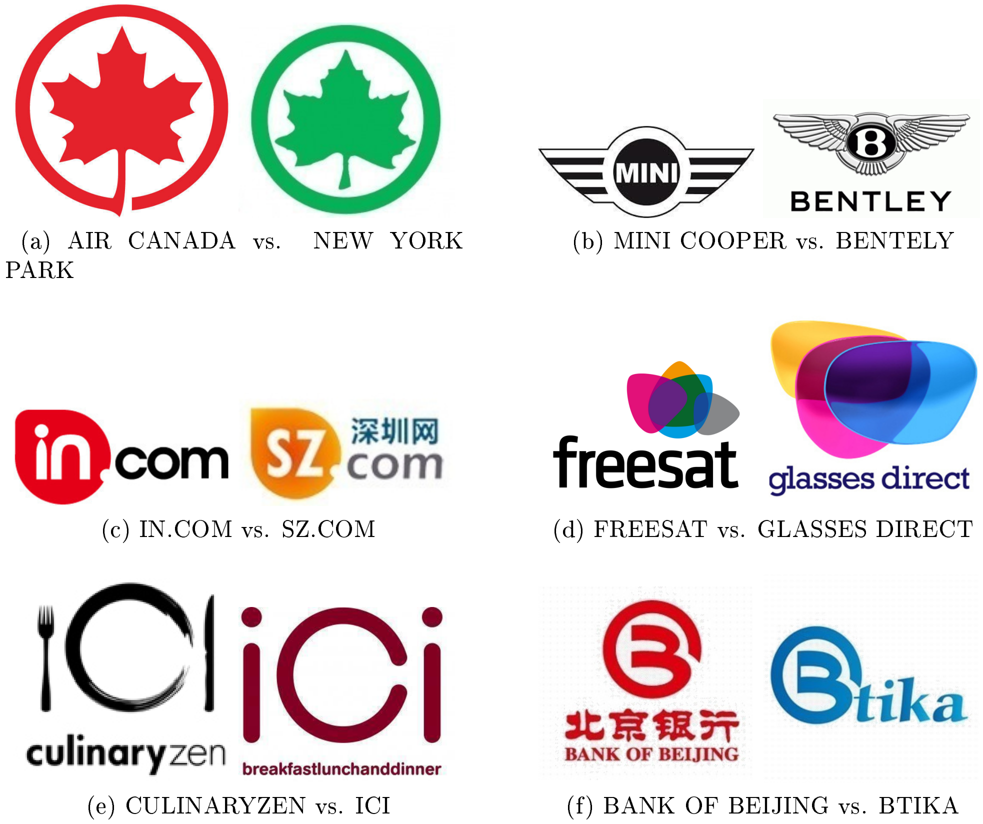
Table 2.1: Sample cases of trademark semantic similarity.

VEGETA	ARGETA
HAWKWOLF	WOLFHAWK
SONY	SQNY
VERSACE	NURSACE

2.1.4 Trademark infringements

Trademark infringement, also known as likelihood of confusion, is a case where there exists similarity between trademarks belonging to different companies. One of the main roles of trademarks is to help consumers make decisions on goods or services, while infringement or similarity will vanish this function of trademarks and lead customers astray. In spite of various attempts taken by the business world and governments to reduce the damage due to trademark infringement, it is still a prominent problem in the commercial world. Especially in the global market, non-negligible organizations intentionally create trademarks similar to well-known trademarks with high likelihood of confusion to take advantage of their reputation on the society.

One of the reasons for increasing trademark infringements is low performance of trademark retrieval in large scale trademark datasets. Another reason is that the amount of trademarks is



(a) AIR CANADA vs. NEW YORK PARK

(b) MINI COOPER vs. BENTELY

(c) IN.COM vs. SZ.COM

(d) FREESAT vs. GLASSES DIRECT

(e) CULINARYZEN vs. ICI

(f) BANK OF BEIJING vs. BTIKA

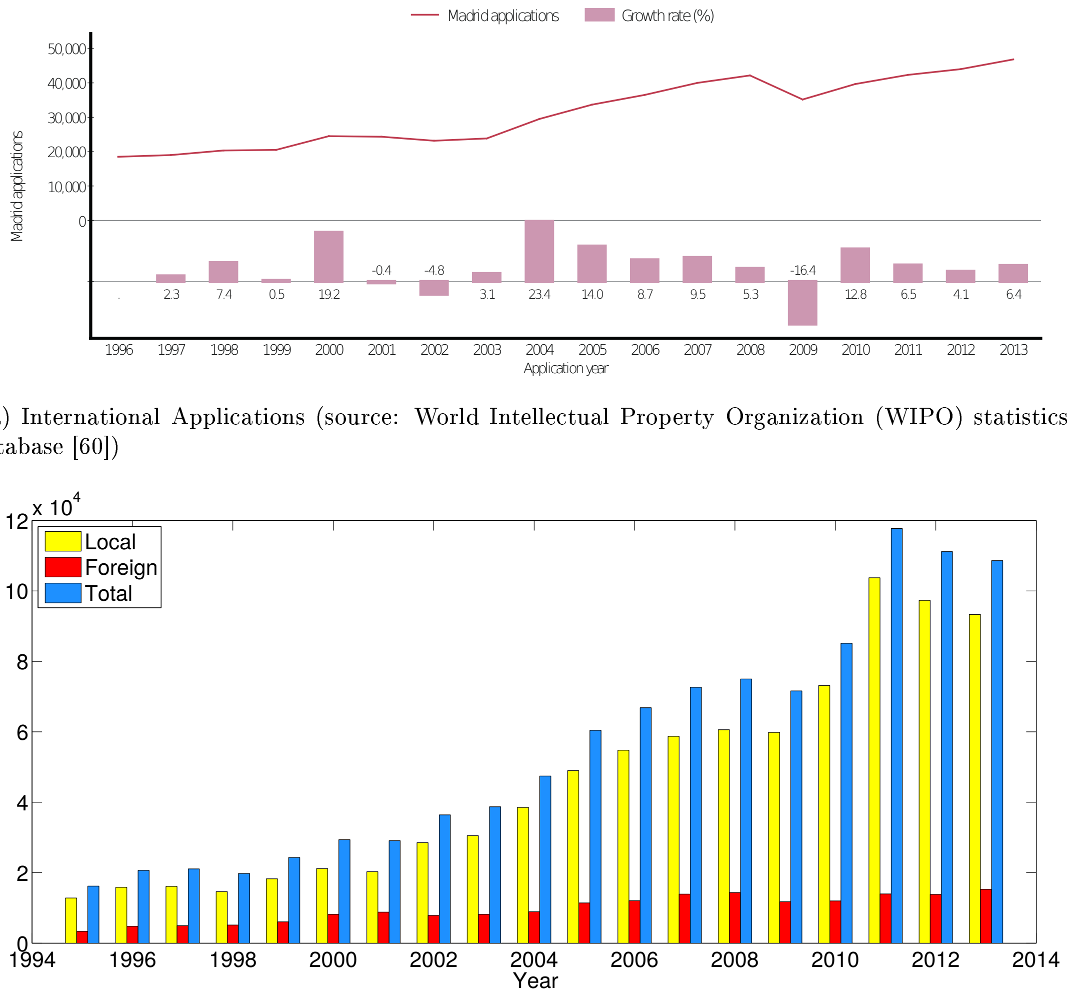
Figure 2.3: Sample cases of trademark visual similarity.

very large and trademarks keep increasing and diversifying. According to the WIPO[3] statistic database [60], in 2012 trademark applications reached about 6.58 million, and in 2013 the number of international applications and registrations were 46,829 and 44,414 respectively, which increased by +6.4% and +5.9%. According to the TPI yearly statistics [34] nearly 1.1 million trademark applications exist, and 10.53% increase was reported in 2013. Figure 2.4 shows in detail the trend of trademark applications.

2.2 Trademark Retrieval Systems

To prevent trademark infringements, government organizations or patent offices offer trademark watch services. In the beginning, these services were based on manual search operation, but high search cost and low retrieval rate (because of exponential increasing of trademarks) have forced watch service providers into looking for alternative solutions. The successful usage of content based image retrieval (CBIR) [79, 84, 91] systems in various retrieval problems promoted applications of special CBIR systems for trademarks, Trademark Retrieval (TR) Systems. A TRS is designed for retrieving visually or semantically similar trademarks from a trademark dataset, which has a trademark similarity search engine based on certain trademark similarity principles. The development of TR systems have enabled semi- or fully-automatized process of trademark registration with low costs and high retrieval rates. Although exponential increase in trademarks and challenging trademark infringements have degraded retrieval

[3] www.wipo.int/



(a) International Applications (source: World Intellectual Property Organization (WIPO) statistics database [60])

(b) Turkey Applications (generated from TPI yearly statistics [34])

Figure 2.4: Statistics of trademark applications worldwide (a) and in Turkey (b).

quality and processing time, it is still the promising method for stopping trademark infringements, due to its successful usage in recent years and potential improvements with promising object recognition techniques.

2.2.1 Manual Annotation-Based TRS

Traditional trademark systems describe contents of trademarks by tagging or coding according to predefined schemes, such as the Vienna classification [59]. The Vienna classification constitutes a hierarchical system that includes 29 categories, like human beings, animals, plants, 145 divisions, 806 main sections, and 903 auxiliary sections (Figure 2.5). The STAR (System for Trademark Archival and Retrieval) system [45] is an example of applying the Vienna classification with users' intervention. Attaching general to specific codes to each trademark with hierarchical schemes like the Vienna classification is considerable in small scale trademark datasets, however, the ever-increasing trademark registration has turned retrieving trademarks by tagging into a tedious, inefficient, high-cost and ineffective process. In addition to the bias of subjective classification [5], lacking of a mechanism to handle newly generated classes [88] and challenge of describing meaningless fractions of trademarks [23] make annotation-based methods impracticable.

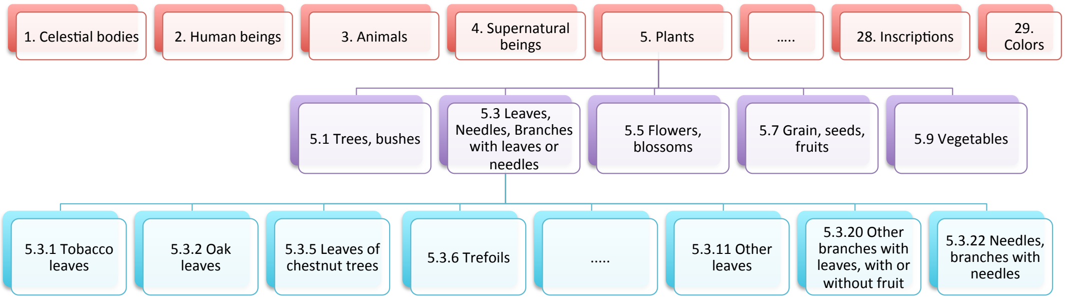


Figure 2.5: Sample part of Vienna classification categories (Adapted from [58]).

Table 2.2: Advantages and disadvantages of Vienna Classification and CBIR based TRS (trademark retrieval systems).

Type	Advantage	Disadvantage
Classification based TRS	Low computational cost Simple retrieval system design	Manual classification bias No adaptation mechanism to new classes Limited classification ability of certain concepts Time consuming
CBIR-based TRS	Fully automatic retrieval No classification bias Adaptability to unpredictable cases Time and cost efficient	Complex retrieval system design High computational cost

2.2.2 Content-based Image Retrieval based TRS

Content-based Image Retrieval (CBIR), also called Query by Image Content (QBIC) and Content-based Visual Information Retrieval (CBVIR), is a system for retrieving images by image content in an image dataset. Before the emergence of CBIR systems, image search used to be similar to the text search, since all dataset images and query images were described with texts. When the two images had common key-words in their description, they would be defined to be similar. However, describing a picture with metadata such as tags, marks and abstract is not enough for high quality search, since “a picture is worth a thousand words”. Instead of using metadata, directly using image content *e.g.*, color, shape and texture is a more desirable method. In Figure 2.6 we depict the main process of CBIR systems. The key-part of this system is feature extraction and feature matching. Therefore, contribution of related studies on CBIR-based TR systems focus on feature extraction and feature matching. In feature extraction part, global feature extraction methods (*i.e.*, edge orientation, moments), local feature extraction methods (*i.e.*, SIFT, SURF) or various combinations of both of global and local features are used. In feature matching part, the query feature will be matched against all features in the dataset, and the implementation of the matching method will be affected by the choice of the features. To be efficient, feature matching is normally accelerated by using a special data structure such as KD-tree [13], hashing mechanism like locality-sensitive hashing (LSH) [33] and inverted index approach [76]. A comparison of advantages and disadvantages of CBIR-based systems versus classification-based systems is summarized in Table 2.2.

In 1992, Kato [41] took the first attempt to retrieve trademarks with a CBIR system. In his TRS, called TRADEMARK, normalized images are mapped to 8×8-pixel grids and a similarity

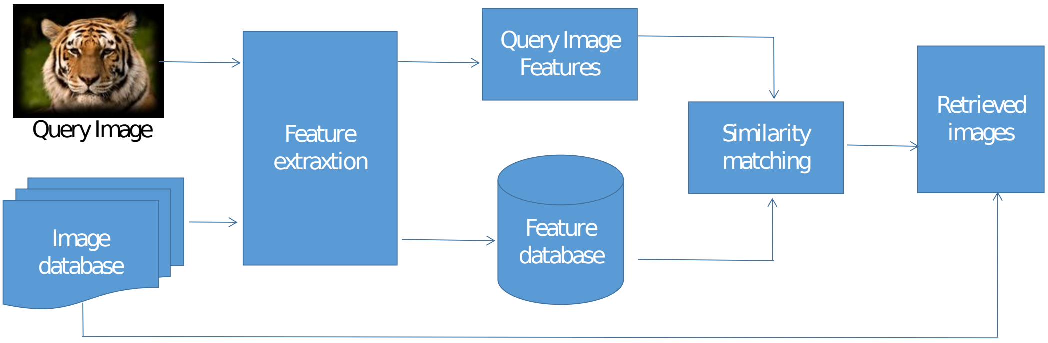


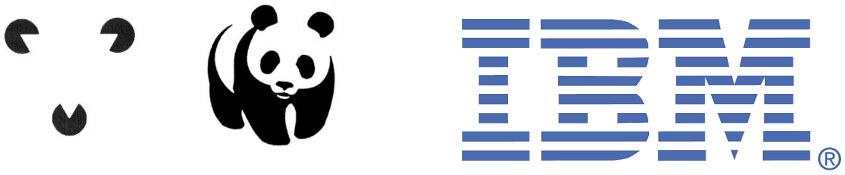
Figure 2.6: Diagram of a CBIR (content-based image retrieval) system.

measure, graphic feature vectors (GF-vector), are obtained by calculating pixel frequency distribution. Relatively advanced shape feature methods used later: Fourier descriptors [22, 31, 88], image moments [18, 22, 36, 88], Zernike moments [43, 86, 90], *simple and low cost shape features* such as aspect ratio [22], circularity [22], etc., Rosin descriptor [22], angular radial transform [22], gray level projection [88],gradient orientation histogram [18, 36], wavelets [18], triangle area representation [3, 4]. A complete list of such studies is provided in Table 2.3. The STAR system [45] is an example of later developed TRS system with shape features in above such as moments, Fourier shape descriptors and gray level projection, besides that it also integrated Vienna classification and text-based image search methods.

In [37], Jiang et al. view aforementioned features as non-geometric features, because their global statistical descriptions of a shape do not contain geometric information. In that case, different images with the same statistical or global feature will be retrieved as similar or similar images with different global or statistical features will be rejected as different. This is the weakness of aforementioned features, but they are useful for preprocessing in large scale TRS systems because of their fast and low-cost attributes. And their retrieve performance could be enhanced by a combination of several of them. However, the combination of multiple features is not advanced than independently using of them according to experiment results of Eakin et al. in [24]. Nevertheless, effectively integrated multiple features [30] will give better retrieval results. In [36], Jain and Vailaya using pipeline integration of multiple features instead of combining similarity results of each individual features.

Early studies show that TR systems describing a whole trademark image with one global feature is not effective. Therefore, segmentation-based methodologies are applied in [5–7, 21, 22, 24, 36, 47]. In these methods, trademark images are segmented to several sub-objects, and each object is encoded by a selected global feature. However, object segmentation is another hard problem which does not always lead to useful segments in an image.

Since human visual system makes use of Gestalt principles for understanding content in images, trademark retrieval systems should also be equipped with the ability to use Gestalt principles. Since 1920s [87], Gestalt psychologists discovered that when people look at a group of objects they try to view them as a whole rather than individually and generate a different perception from the simple sum of objects. We can see a white triangle bitten by three Pac-Man and panda in Figure 2.7a and we can recognize an IBM logo from its parts in Figure 2.7b. However, without perceptual organization they are just black, closed regions and blue-bold segments. Gestalt psychologists argue that human visual system organizes objects according to certain principles: similarity, continuation, closeness, proximity and so on. Eakins et al. [21, 22, 24],



(a) Gestalt triangle and panda (b) IBM logo with Gestalt design

Figure 2.7: Example of for how Gestalt principles effect trademark perception.

Alwis et al. [5, 7] and Jiang et al. [37] have integrated Gestalt principles to their TRS. The ARTISAN system [25] developed by Eakins is an example of TR systems which apply Gestalt principle for trademark retrieval. In [37] Jiang et al. extract Gestalt elements such as circles (arcs), parallel lines, concentric circles, polygons from trademark images, then find similar trademarks with weighted bipartite graph (WBG).

In addition to shape features, color is also essential for TR. In the early studies, because of binary image datasets and its other disadvantages, color features were not considered while evaluating similarity. While several studies [45, 62, 70] included color in TR, their and our results show that color also yields good retrieval performance in special cases.

Although global features give the whole outline of trademark images or objects of trademark images, they lose the significant local information in them. Paijit [44] investigated trademark image retrieval by local features. He pointed out detecting partial similarity of trademarks is better than the matching ability of global features. For example, he suggested that local features such as SIFT is robust to partial matching. In his TR systems, he uses local features and imitates human visual perceptual judgment by relevance feedback and decision tree classification.

Table 2.3: Shape-based trademark retrieval methods in the literature.

Group	**Approach**	**Study**
	Fourier descriptors	[22, 31, 88]
	Moment variants	[18, 22, 36, 88]
	Zernike moments	[43, 86, 90]
Simple and Low-Cost Shape Features	Aspect ratio	[22]
	Circularity	[22, 86]
	Convexity	[22, 90]
	Compactness	[90]
	Eccentricity	[3, 90]
	Distance to centroid	[86]
	Rosin descriptor (triangularity, rectangularity and ellipticity)	[22]
	Triangle area representation (TAR)	[3, 4]
	Angular radial transform	[22]
	Gray level projection	[88]
	Gradient orientation histogram (edge direction)	[18, 36]
	wavelets	[18]
	Shape-context	[71]

2.2.3 Summary

The importance of trademarks makes them necessary, valuable and significant in the business market and increases their applications and registrations exponentially. Most trademark datasets at patent offices contain millions of trademarks with different types. This turns the manual trademark retrieval into a laborious, burdensome and challenging problem. The emergence and successful applications of CBIR systems have lead studies on TR systems to using CBIR-based systems for trademark retrieval. Although studies on CBIR systems are promising solutions for TR systems, the trademark retrieval problem has particular aspects: Directly employing methods from CBIR systems or other methods do not yield desired results.

CHAPTER 3

FEATURE EXTRACTION FROM TRADEMARK IMAGES

In this chapter, we briefly introduce various descriptors widely used in image retrieval. Moreover, we describe the ones implemented and compared in this work in more detail.

3.1 Classification of Descriptors

Trademark visual similarity could be achieved by comparing color, shape and texture elements of images. From this aspect, descriptors could be divided into *shape-based descriptors*, *color-based descriptors* and *texture-based descriptors*. In addition, according to the scales of the described regions, they could be dived into *local descriptors* and *global descriptors*. Moreover, as suggested by Jiang et al. [37], features can be non-geometric descriptors (*e.g.*, edge direction histogram and moments), which are global and statistical descriptors with non-geometric information, and geometric descriptors (*e.g.*, shape context [11]).

3.2 Shape Descriptors

Shape is an important source of information for describing content in images. The survey conducted by Her et al. [30] showed that shape, color and pronunciation of trademarks make great impressions to the customers, while shape features are considered to be the most significant criteria for determining similarity degree. Shapes of images bear semantic meaning and might express the main intentions of trademarks.

A single shape descriptor might not be adequate to comprehensively describe shapes since each of them has its advantages and disadvantages. Therefore, most CBIR systems integrate several diverse shape descriptors. To build a well-designed TR system, we should use descriptors depending on needs, conditions, efficiency and effectiveness. Yang et al. [89] carried out a detailed study on the existing shape-based feature extraction methods. In their study, they pointed out the essential properties of efficient shape features, which are given and explained in Table 3.1.

We can classify shape analysis methods from various aspects [48, 61, 89, 91]:

- The first classification is *contour-based* and *region-based methods*. The bigger difference of these two approaches is that the former just considers objects' external characteristics

Table 3.1: Essential properties of efficient shape analysis method (Source: [89]).

Properties	**Explanation**
Identifiability	Shapes which are found perceptually similar by human have the same feature different from the others.
Translation, rotation and scale invariance	The location, rotation and scaling changing of the shape must not affect the extracted features.
Affine invariance	The affine transformation performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the ”straightness” and ”parallelism” of lines. Affine transformation can be constructed using a sequence of translation, scales, flips rotations and shears. The extracted features must be as invariant as possible with affine transformation.
Noise resistance	Features must be as robust as possible against noise, i.e., they must be the same whichever be the strength of the noise in a give range that affects the pattern.
Occlusion invariance	When some parts of a shape are occluded by other objects, the feature of the remaining part must not change compared to the original shape.
Statistically independent	Two features must be statistically independent. This represents compactness of the representation.
Reliable	As long as one deals with the same pattern, the extracted features must remain the same.

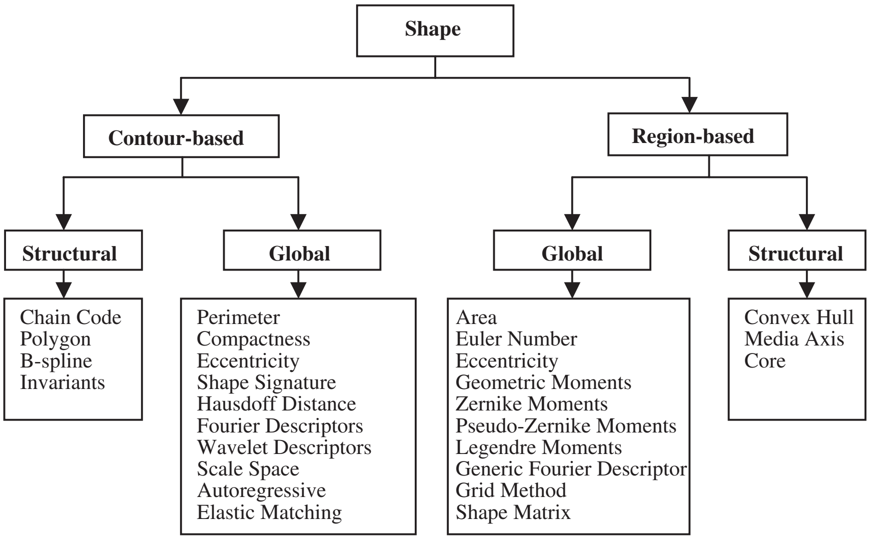


Figure 3.1: Classification of contour-based and region-based shape representation techniques (Adapted from [91]).

(*i.e.*, the object boundary), while the latter type extracts shape features from the the region. The methods in each class are also further divided to sub-classes like *structural approaches* or *global approaches*. In global approaches, the shape is represented as a whole, while, in structural approaches, the images are manually or automatically segmented into several objects, and the shape descriptor of the image is a combination of all shape descriptors of these segments. See Figure 3.1 for an illustration of these two categories.

- In the second classification, the methods are dived into *space domain* and *transform domain*, which is decided by whether the features are extracted from the spatial domain or the spectral domain.

- The third classification of shape analysis methods is based on information preservation; *information preserving* (IP) and *non-information preserving* (NIP). With the IP methods shapes can be accurately reconstructed from its descriptor, while the NIP methods are only capable of partial reconstruction. For large-scale TR systems, both kinds of shape analysis methods are necessary: NIP methods for fast pruning-based search, while the IP methods for detailed search. In hybrid TR systems, these methods could be combined in different steps.

Although shapes are critical for trademark similarity, trademarks can include complex patterns, which might not be captured by shape only, and other visual cues should be used in conjunction.

3.2.1 Shape context

Belongie et al. [11] developed a novel shape descriptor called shape context. A shape context of a point is defined as the spatial distribution of other points on the same shape to that point. By calculating a distance between two shape contexts, one can find corresponding points in two shapes, and then deform one shape to look similar to another shape. The cost of deformation is then taken as the similarity of two shapes.

The shape context descriptor is translation and scale invariant, and could be rotation invariant by a minor extension. Shape context if used together with log-polar histogram is invariant to small affine distortion owing to pose changes and intra-category variation. It is also robust to noise and occlusion [12].

To obtain the shape context of a shape, the first step is to uniformly sample points from inner and outer outline of the shape. Well-distributed sample points on the outline of a shape are approximate representation of a shape. By applying edge detectors like Canny [15] and then uniform sampling, we get n sample points P:

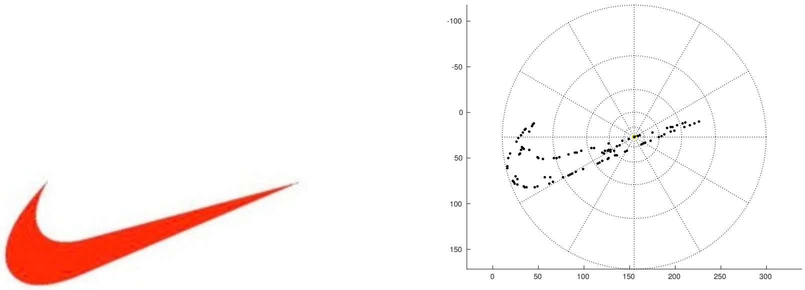
$$\mathcal{P} = \{p_1, p_2, \dots, p_n\}, p_i \in \mathbf{R}^2. \tag{3.1}$$

The second step is to connect each point with its $(n-1)$ neighbors, and generate $(n-1)$ vectors for each point to express the configuration of the entire shape. For the whole shape, we use $n \times (n-1)$ vectors, which form a rich description of the shape. For the sake of efficiency, studies often approximately describe $n-1$ related vectors of each sample point by describing distribution of relative $n-1$ point positions with diagram of log-polar histogram bins. In original work and our implementation, 5 radius bins θ and 12 angle bins $logr$ is chosen for log polar histogram. In that case, all $n \times (n-1)$ vectors could be replaced by n histogram of 60 bins. This is a robust and compact descriptor.

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\}. \tag{3.2}$$

See Figure 3.3 for polar histogram of a sampling point on a sample logo.

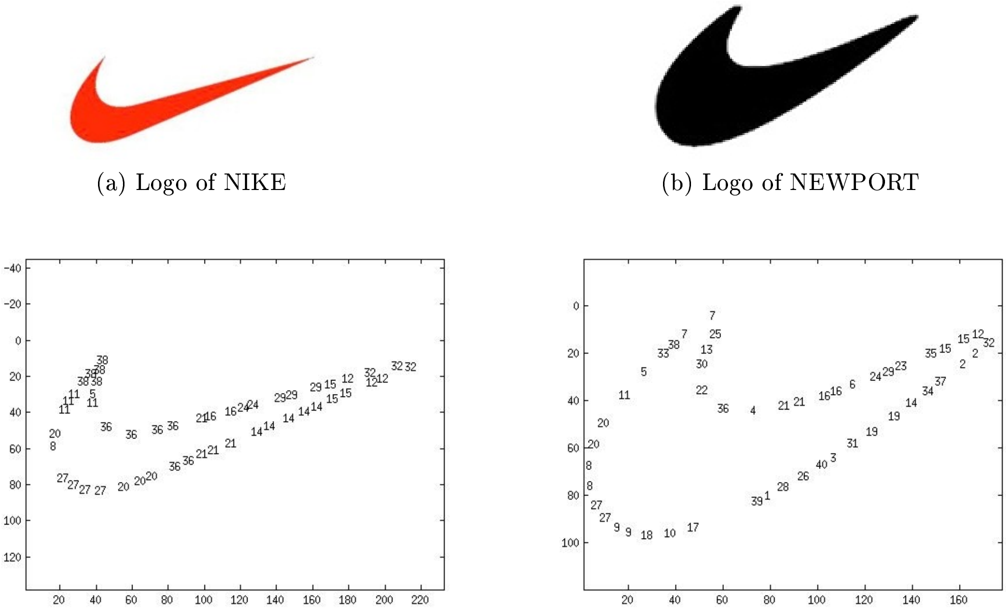
In Belongie et al. [12], the total matching cost of corresponding points is considered as the similarity level between two shape context descriptor. Since a point's shape context distribution is represented as a histogram, χ^2 statistical test is used for calculating matching cost of



(a) Logo of NIKE (b) a polar histogram of a point on NIKE logo

Figure 3.2: A polar histogram of a sample logo.

points' shape contexts. For finding corresponding points, they use the Hungarian method, a computationally expensive method since it is $O(n^3)$ [12].



(a) Logo of NIKE

(b) Logo of NEWPORT

(c) Shape-context of NIKE logo

(d) Shape-context of NEWPORT logo

Figure 3.3: Shapeme of sample logos.

To improve efficiency of shape context, Mori et al. [52] introduced two approaches; one is the representative shape-context method, another is the shapemes method. In our study, we use shapemes, and skip the discussion on representative shape context here. In shapemes method, vector quantization with K-means classification is applied on shape context descriptor vectors (shapemes method is shape context method plus bag of visual words (BoVW)). In Figure 3.3, we see that the logo of NIKE and NEWPORT both have similar shape, therefore they should have common context points. In Figure 3.3a and 3.3b, we see that they have common points such as 36, 11, 12 at corresponding positions.

3.2.2 Gradient orientation histogram

Intensity gradients of images are also very informative as shown by [18, 36]. In Figure 3.4, we observe that similar images show similarity in gradient orientation histograms. To obtain gradient orientation histogram vector, we first applied Canny edge detector [15] to the median-filtered image and calculate orientation of each gradient point. From the quantized orientation of each gradient point, we create a gradient orientation histogram. The quantization level of gradient points is essential, because too fine quantization is very sensitive to rotation, and too coarse quantization leads to a lack of distinctiveness. For comparing two gradient orientation histograms, we used Euclidean distance.

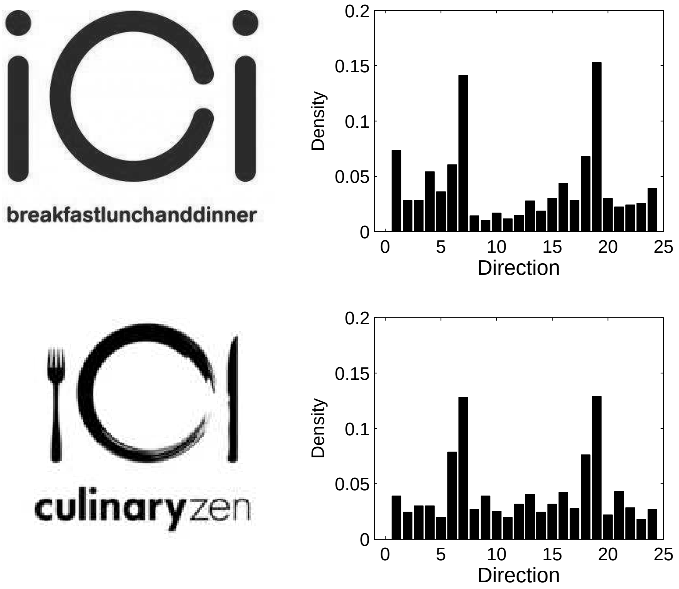


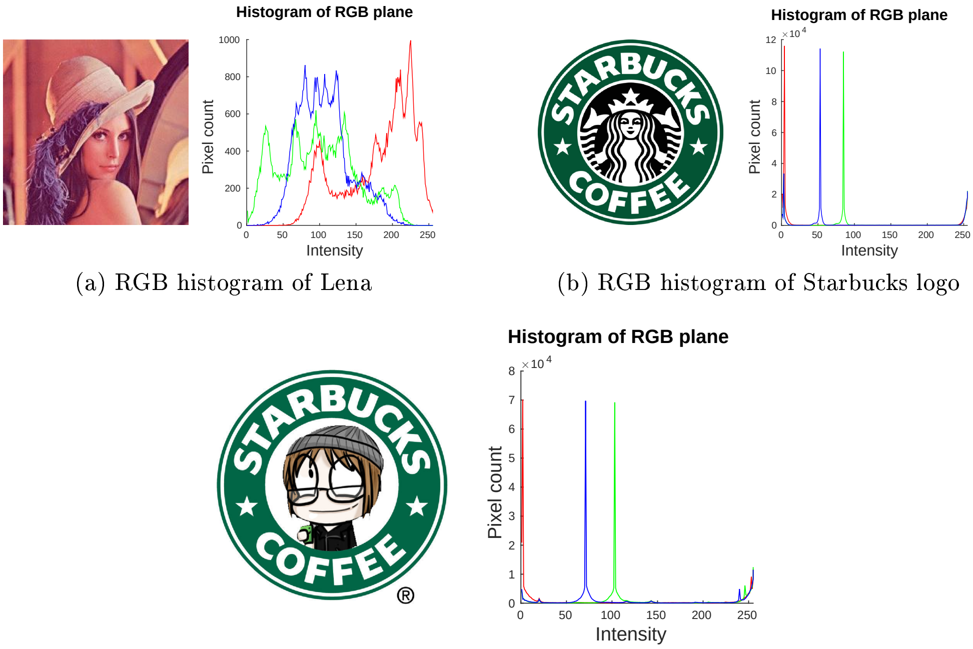
Figure 3.4: Edge gradient orientation histograms of sample logos.

3.3 Color Descriptors

Color is an elementary visual element and practical in trademark retrieval. Color is a key element for creating a particular, attractive and distinctive logo. Her et al. [30] suggest that color of trademarks leave a significant impression on customers. Moreover, Kesidis et al. [42] mention that color schemes should be additionally registered when applicants define color as a feature of the trademark. In their example, color schemes are registered by generating code or text descriptions. Therefore, if a trademark own a particular color scheme, the retrieval process should include color scheme retrieval.

Image retrieval by color is popular in CBIR systems [35, 54], because of its simplicity, efficiency and effectiveness. Mostly, histograms of colors of pixels are used as color-describing features. As Figures 3.5b and 3.5c show, if the original images have similar color schemes, then their histograms are also similar to each other. Color histogram method is translation and rotation invariant, and it could achieve scale invariance with proper normalization.

However, retrieval with color in TR systems will not be as effective as in CBIR systems. Be-



(a) RGB histogram of Lena

(b) RGB histogram of Starbucks logo

(c) RGB histogram of fake Starbucks logo

Figure 3.5: Examples of RGB color histograms.

cause color schemes of trademarks are plain compared to other image types such as paintings, photographs, etc. And they do not contain enough color information for being distinctive. For example, in Figure 3.5 we can notice that the RGB histogram of Lena has obvious variety, while RGB histogram of two Starbucks logos are comparatively simple. Therefore in most TR systems and related applications [45, 62, 70], color features are used together with shape and texture features.

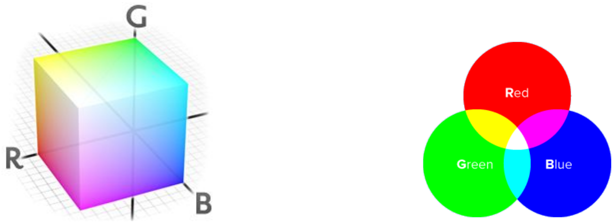
The retrieval performance of color histogram methods depends on several aspects: 1. **Color space:** RGB, CMYK, CIELUV, YIQ, YPbPr, xvYcc, HSV, HSL, etc. 2. **Normalization:** To achieve scale invariance, the histogram should be normalized properly. 3. **Distance Metrics:** The similarity of histograms are calculated by using certain distance measures. See Appendix B for various normalization methods and distance measures tested in the thesis.

In this work, we have tested RGB and HSV color histograms using different quantization, normalization and distance metrics. In the following, we will introduce them in more detail.

3.3.1 RGB color histogram

The RGB color histogram captures the distribution of color in the RGB color space. The RGB color model, where RGB stands for red, green, blue, is an additive color model. Mixing red, green, and blue light together in various ways reproduces bunch of colors. Figure 3.6 shows cube and adding model of RGB color space. RGB color is the most popular color model and easy to quantize:

$$I = V \times \frac{N}{255}, \tag{3.3}$$



(a) Coordinate model (Adapt from [1])

(b) Adding model Adapt from [80])

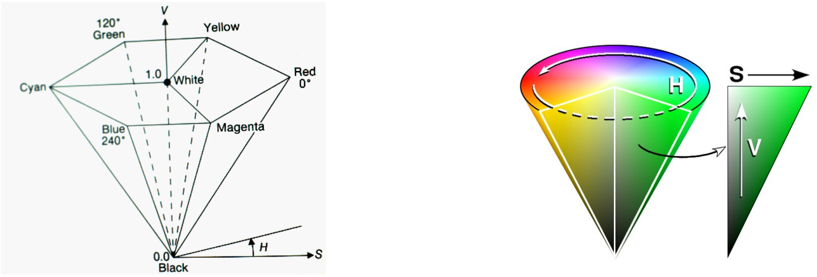
Figure 3.6: RGB color model.

where V is the value of R, G, B channels, N is quantization parameter, I is the quantized level of R, G, B values. The quantization parameter should be neither too fine nor too coarse, because of efficiency and accuracy concerns. For a balance and convenient quantization, we set up quantization parameter of each channel as 4 and 8. Choose quantization parameter among divisor of 256 will make quantization uniform and convenient. This yields $4^3 = 64$-bin and $8^3 = 512$-bin RGB color histograms.

3.3.2 HSV color histogram

Compared to the RGB color space, the HSV (Hue, Saturation, and Value) color space simulates color perception of human more properly. If we look at the cube model of RGB color space in Figure 3.6a, we could discover that some close colors in RGB space are very different at human perception. While the Euclidean distance of colors in HSV cylindrical color model could reflect the similarity degree of these colors at human perception precisely.

HSV (Hue, Saturation, and Value) is a cylindrical color model, where Hue ranges from 0 to 360, and Saturation and Value ranges from 0 to 1 - see Figure 3.7a. From 3.7b we can observe Hue panel is segmented into various color regions, and the higher saturation value the more gray in color, and the conjunction of Value value and Saturation value decides the intensity or brightness of the color.

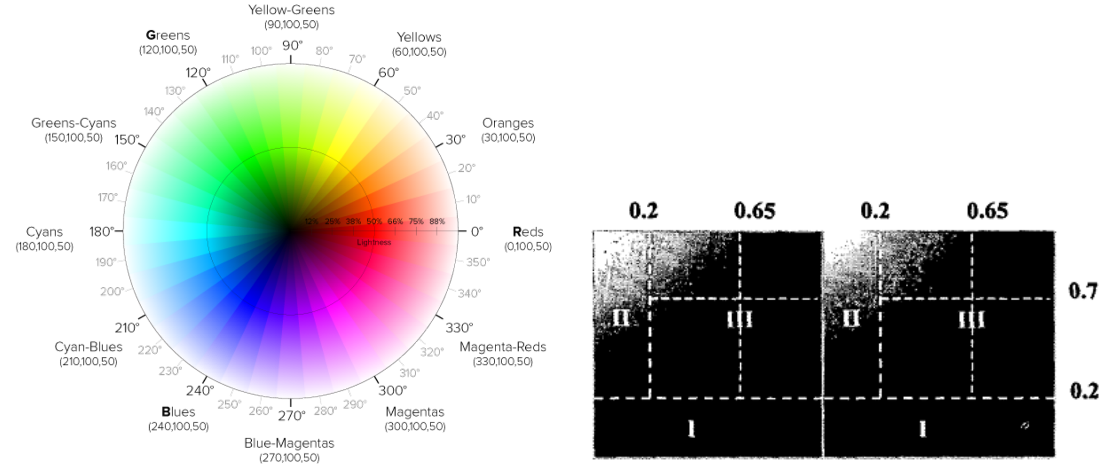


(a) HSV coordinate system (Adapt from [65])

(b) Illusinary of HSV (Adapt from [85])

Figure 3.7: HSV color model.

The RGB color model is a cubic model, so we quantize R, G, B channels uniformly. Nevertheless, it is not appropriate that quantize cylindrical HSV model uniformly. That is because the color distribution in HSV color model is not uniform. In Figure 3.8a, we can see that the range of colors displayed in Hue panel such as red, orange, yellow, green, cyan, blue and purple are not uniform, and three primitive color red, green, blue occupy main part of the panel. In addition to that, Saturation and Value panel on Figure 3.8b are also divided three non-uniform regions based on human perspective. Zhang et al. [46] and Li et al. [28] developed 36 and 72 bins HSV color histogram method by non-uniformly quantizing Hue, Saturation and Value channel of HSV color model. Their experiment results shows their methods are superior to 166 bins HSV model developed by Smith et al. [78].



(a) Hue panel (Adapt from [80])

(b) SV panel quantization (Adapt from [46])

Figure 3.8: Non-uniform quantization of H,S,V panels.

3.4 Texture Descriptors

Texture is basically a repetitive or non-repetitive arrangement of intensities [75]. Since they can be very distinctive, texture features are commonly used in CBIR systems [26, 32].

Local binary patterns (LPB) [56, 57] is a simple, efficient method for extracting textural information. In LBP, a structural pattern is extracted from each pixel of image by comparing its intensity with its N neighbors in a certain radius. For convenience, we attach an order id form 1 to N to each neighbor of center pixels in clock-wise or anti-clock-wise: The id of neighbor at starting point is 1, and the next neighbor is 2 and so on. If the intensity value of a neighbor is bigger than or equal to the intensity of the center pixel, the comparison result is 1, otherwise it is 0. After comparison, a binary pattern string is created by concatenating the comparison results of neighbors by order id. The decimal value of this binary string is the type number of pattern or spatial structure. By repeating the above process for each pixel of the image, the spatial structure of each image pixel is obtained. After counting the occurrence of each pattern in an image, the resulting LBP feature vector is obtained as a 2^n-bin vector, where the i^{th} bin represents the occurrence of the i^{th} pattern. For example, if we were to conduct the comparison with 8 neighbors, we would obtain a vector of 256 bins, where each bin would represent the occurrence of the related pattern. Ojala et al. in [56] use unique $LBP_{P,R}$ numbers to characterize the spatial structure of the local image texture,

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^P, \tag{3.4}$$

where P is the number of neighbors in a circle with radius R, and g_p is the intensity (gray value) of pixel p, g_c is the intensity of the center pixel, and $s(x)$ is defined as follows:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}. \tag{3.5}$$

The simple version of LBP is neither rotation invariant nor robust to spatial resolution. To achieve rotation invariance, the operator in Equation 3.6 is used for removing the effect of rotation:

$$LBP_{P,R}^{ri} = \min \{ROR(LBP_{P,R}, i) \mid i = 0, 1, \ldots, P-1\}. \tag{3.6}$$

where $ROR(x, i)$ performs a circular bit-wise right shift on the P-bit number x for i times. Each normal LBP pattern have 8 different patterns (including itself). Therefore, when we just consider the 8 neighbor format, $256/8 = 32$ rotation invariant LBP patterns exist.

The experimental results of Ojala et al. [57] suggest that rotation invariant LBP patterns have less discrimination ability. Their results also show that certain kind of patterns, named 'Uniform' patterns by the authors, are fundamental patterns in textures. The definition of Uniform pattern is given in Equation 3.8. If $R = 1$ and $P = 8$, there exist 58 Uniform patterns and other patterns are categorized into not uniform patterns. In other words, we can get a 59-bin LBP histogram by applying this Uniform LBP operator as follows:

$$LBP_{P,R}^{u2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U(N(P,R)) \leq 2 \\ P+1, & \text{otherwise} \end{cases}, \tag{3.7}$$

where $s(x)$ is as defined in Equation 3.5 and $U(N(P,R))$ is defined as follows:

$$U(N(P,R)) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|. \tag{3.8}$$

To create rotation invariant and Uniform LBP, the rotation invariance operator is applied on $LBP_{P,r}^{u2}$ in Equation 3.6 as follows:

$$LBP_{P,R}^{riu2} = min\left\{ROR(LBP_{P,R}^{u2}, i) | i = 0, 1, \ldots, P-1\right\}. \tag{3.9}$$

58 Uniform patterns own 9 rotation invariant patterns, therefore, LBP patterns are categorized into 9 rotation invariant Uniform patterns and one that is not.

3.5 Keypoint-based Descriptors

Not all pixels carry distinctive and informative content that can be used for matching, and therefore, one might consider extracting features only at key locations, called keypoints, that are more likely to contain useful content.

3.5.1 Features from Accelerated Segment Test (FAST) detector

FAST is a real-time corner detector proposed by Rosten and Drummond [68]. Due to its computational efficiency, it is used with interest point descriptors. The FAST method will be used with the descriptors in our work, and therefore, we explain it first in this section.

To identify whether a pixel p is an interest point, the FAST corner detector compares the brightness value of this center pixel with the set of 16 pixels, S, i.e., the pixels within a 3-pixel distance. The pixel p is deemed interesting if the brightness of p and the pixel set S meet the following requirement:

- A set of N contiguous pixels $s \in S$, $\forall x \in s$, the intensity of pixel x, $I(x)$, and the intensity of pixel p, I_p, meet the condition that makes $(I_x) > I_p + t$ *or* $(I_x) < I_p - t$, where t is a threshold and default N is 12.

To accelerate the detection, a high-speed test is employed in FAST. In this test, non-key points are rejected by examining four preselected pixels from 16 pixels when N is equal to 12. Assume we labeled these 16 pixels from 1 to 16, then these pixels are no. 1, 9, 5 and 13 pixels. First, we check whether brightness of pixels 1 and 9 lie in $[I_p - t, I_p + t]$, where I_p is the brightness value of the center pixel p and t is the threshold. If they do, the pixel p is not a corner. Otherwise, pixels 5 and 13 are further examined to check whether their brightness values are bigger than $I_p + t$ or smaller than $I_p - t$. If any three pixels among no. 1, 9, 5 and 12 pixels all brighter than $I_p + t$ or darker than $I_p - t$, the rest of pixels will be examined for the final decision. On the average, Most non-corner points of images will be rejected in a few steps. As a consequence, above procedure can boost up the speed of detecting corner points by reducing number of redundant checking.

However, this high-speed test become less efficient and effective when we changed the optimal running parameters. To make it more robust, the author applies machine learning and non-maximum suppression.

3.5.2 Scale Invariant Feature Transformation (SIFT) keypoint detector

SIFT [50] is a scale invariant key-point detector and descriptor proposed by Lowe in 1999. It has been successfully applied in various vision problems such as object detection and recognition, image stitching, 3D modeling. It detects interest points/key-points and describes them with a rich and robust rotation, translation and uniform scale invariant descriptor. In addition, it is also partially overcome affine distortions, illumination changes and noises. The SIFT algorithm is mainly accomplished in the following four steps:

Scale-space extrema detection

As we mentioned in Section 3.5.1, key-points are mainly found around corner points. However, early corner detectors, such as Harris [29], are not scale invariant though they are rotation invariant. To find stable and scale-invariant key-points of images, scale-space filtering LoG (Laplacian of Gaussian) is employed with various σ values, where σ is a scaling parameter. In this process, LoG function can detect various-scale key-points shown in Figure 3.9a. These circle regions are saved in a list of (x, y, σ), where the center of a circular image region, (x, y), is the location of a key-point and the σ value is the scale at which this key-point is found. To be more efficient, above procedure could be approximated by subtracting two identical images filtered with two Gaussian kernels with close scales, such as σ, $k\sigma$, $k^2\sigma$, $\ldots$, $k^(n-1)\sigma$ (k is the constant number and n is number of scales sampled in each octave):

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma), \end{aligned} \tag{3.10}$$

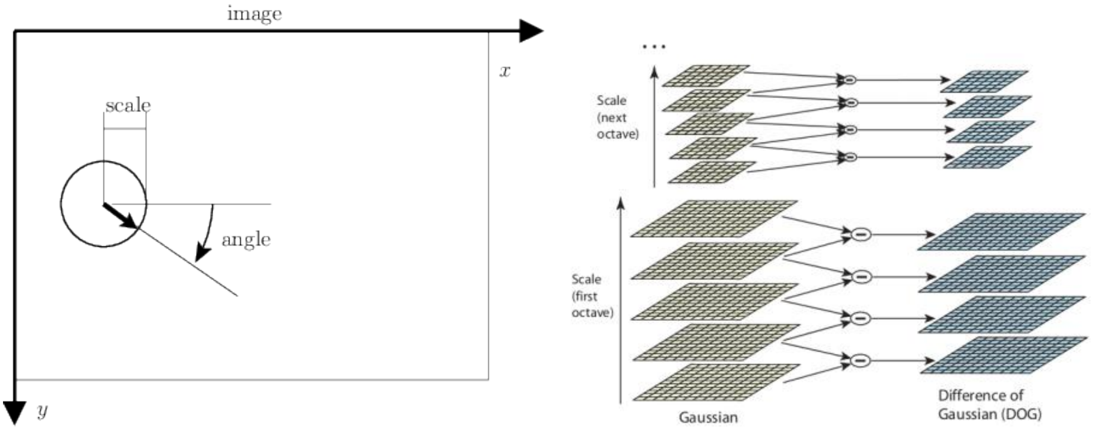
where $L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$ with the Gaussian kernel $G(x, y, k\sigma)$ at scale $k\sigma$ as follows:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y), \tag{3.11}$$

where $G()$ is the Gaussian with standard deviation σ:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma}. \tag{3.12}$$

Candidate key-points are pixels whose brightness values are the lowest or the highest of its eight neighbors at the same scale and nine corresponding neighboring pixels of each of the upper and lower neighboring scales.



(a) SIFT keypoints are circular image regions with an orientation (taken from [83])

(b) The computation of the Difference-of-Gaussian image pyramid (taken from [50])

Figure 3.9: Explanation of the first step of SIFT implementation.

Keypoint localization

Locations of candidate key-points are found in the previous step, but they are not very accurate. To improve positioning, Taylor series expansion of scale space is implemented. Then the candidate points that comprise low contrast or edge responses are eliminated to enhance the matching stability.

Orientation assignment

To achieve rotation invariance, SIFT assigns a main orientation to each key-point descriptor. The key-point orientation is calculated by creating orientation histogram with gradient magnitude and the direction of neighborhood pixels in the blob region of this key-point. The gradient magnitude m and direction θ of neighborhood pixels are computed as:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}, \tag{3.13}$$

$$\theta(x, y) = tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y))), \tag{3.14}$$

where $m(x, y)$ and $\theta(x, y)$ are the magnitude and gradient directions of pixel at (x, y), and $L(x, y)$ is its gray level.

The orientation histogram has n bins (n is 36 in the original work). Instead of using the highest bin as an orientation, Lowe chose directions of bins, which is above the 80% of the highest bin as orientations of this key-point; in other words, a key-point could own multiple orientations there does not exist an extremely obvious direction.

Key-point descriptor
At this step, location, orientation and scale information of key-points have been extracted. Now, we describe key-points with gradient distribution of its neighbors within the detected region. As an example, the pixels in the 16×16 neighborhood surround the key-point are taken and separated into 16 sub-blocks, each of which is 4×4 pixel^2. An 8-bin gradient orientation histogram is calculated in each sub-block. Finally, concatenating orientation histograms from each sub-block gives a 128-bin key-point descriptor.

Key-point matching
Key-points from two images are matched by calculating a distance metric like Euclidean distance. For efficiency, Lowe approximated Euclidean distance by the sum of vectors' dot product. To reject false matches, the matching cost of the most similar descriptor should be higher than the similarities to other descriptors by a threshold Γ. Normally the Γ is 0.8. According to original study, this procedure eliminates 90% false matches, but sacrifices 5% correct matches.

3.5.3 Triangular SIFT

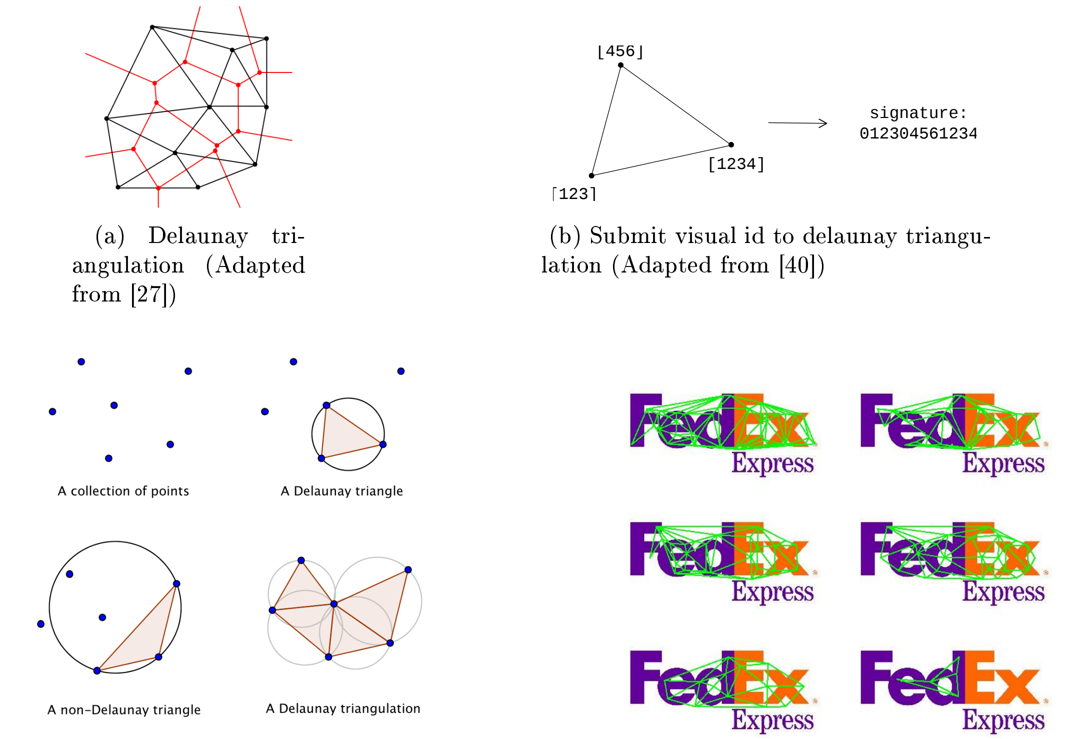
SIFT is an effective, stable and robust descriptor but it is not efficient or scalable, especially in a large scale dataset. To scale up the SIFT descriptor, the local geometry information must be incorporated in new descriptors. In [40], as Figure 3.10d shows, SIFT features whose scales lie on certain scale range are grouped as triples applying Delaunary triangulation. Each group is represented by union of signatures of SIFT key-points at vertex of triplets (see Figure 3.10b).

As Figure 3.10c shows, any circumscribed circle of Delaunay triangulations $DT(P)$ of a set of points P in a plane does not include any vertex points of other Delaunay triangles inside of it. Valuable properties of Delaunay triangulations: (i) $DT(P)$ is unique for P. (ii) It is robust to outliers when features are grouped by triplets. Due to these properties we could find matched triplet SIFT groups in similar images. Group matching of key-points are superior to individual matching when distinctiveness is not adequate. For instance, indexing 500 million features with 10,000 indicies should be considered as not distinctive, while more fine indexing requires more computation power and time for feature classification. With employing Delaunay triangulations, grouping 1,000 distinctive SIFT features leads to 1 billion triplet SIFT groups.

3.5.4 Speeded Up Robust Features (SURF)

The SURF algorithm by Herbert et al. [10] is, in principle, similar to SIFT, but it is faster, and it shows better performance than SIFT in certain cases. After having introduced SIFT in detail, it suffices to describe SURF by comparing it with SIFT.

Interest point detection
To be scale invariant, SIFT applies a series of Laplacian of Gaussian (LoG) on the image, and approximates it with Differences of Gaussians (DoG). Although the DoG operation speeds up the computing, it is still slow. Instead of using DoG, SURF uses convolutions of different-size box filters with integral image [10] to find interest points. More precisely, SURF applies a



(a) Delaunay triangulation (Adapted from [27])

(b) Submit visual id to delaunay triangulation (Adapted from [40])

(c) Non-Delaunay and delaunay triangulation (Adapted from [51])

(d) Scaled triangulation-SIFT on Fedex logo (Adapted from [40])

Figure 3.10: Delaunay triangulation and its implementation with SIFT.

Hessian-matrix-based blob detector to find interest points. The Hessian matrix $H(p, \sigma)$ at point $p = (x, y)$ at scale σ is defined as:

$$H(p, \sigma) = \begin{pmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{pmatrix}, \tag{3.15}$$

where $L_{xx}(p, \sigma)$ are $L_{xy}(p, \sigma)$ are defined as follows:

$$L_{xx}(p, \sigma) = I(p) * \frac{\partial^2}{\partial x^2} G(\sigma), \tag{3.16}$$

$$L_{xy}(p, \sigma) = I(p) * \frac{\partial^2}{\partial xy} G(\sigma), \tag{3.17}$$

where $G(\sigma)$ is the Gaussian kernel at scale σ.

Orientation assignment

Rotation invariance is achieved by calculating the orientation of key-points. SIFT generates gradient direction histogram to find the main direction, while SURF uses Haar wavelet responses in both horizontal and vertical directions within a circle of radius 6σ, where σ is the detected scale of interest point.

Key-point descriptor

A detected key-point is described with the wavelet responses of 4×4 orientated square subregions in the region of key-point. Orientation of sub-square regions is obtained at the last step, and 5×5 points are sampled in each sub-square. For each sub-square, the sums of dx, dy, $|dx|$ and $|dy|$ are computed relatively to the orientated direction.

3.5.5 Histogram of Oriented Gradients (HOG)

The HOG [19] is another well-known local feature descriptor widely used for object detection and recognition (*e.g.*, pedestrian detection) in computer vision and pattern recognition. It shares similar principles with SIFT and gradient orientation histogram methods, though, it is different from them in several aspects.

HOG is a window-based descriptor, and it does not have a detector defined as part of the method. Generally, HOG is used with key-point detectors such as FAST or sampling techniques to select key points. In HOG, an n grid of cells is placed on a window. For each cell in the window, a gradient orientation method, discussed in Section 3.2.2, is applied. As a consequence, each cell is described by a q-bin vector, and the key-point is the concatenation of n^2 many q-bin histograms.

In our HOG implementation, we chose FAST corner detector as our key-point detector, by taking the top 100 stable key-points from a given image. For each key-point, we applied a window with 2×2 cells. Then 9-bin gradient orientation histograms are extracted form each cell. By concatenating 4 histograms, a 36-bin HOG feature vector is generated for each key-point.

3.5.6 Oriented FAST and Rotated BRIEF (ORB)

In [69], Rublee et al. proposed ORB feature descriptor, an efficient alternative to SIFT. It is a very fast binary descriptor based on FAST detector and BRIEF [14] binary descriptor. It is also invariant to rotation, and it is robust against noise. It has been successfully used in many computer vision problems such as object recognition and 3D reconstructing, especially, on embedded devices such as mobile-phones. The most appealing part of the ORB methods is its memory and time efficiency. The feature extraction process of ORB is very FAST, for instance, ORB features of million images could be extracted in 10 minutes with multi-threaded computing, while SIFT and SURF take around 24 hours. Moreover, since it uses Hamming distance, matching is also very fast.

The ORB algorithm involves the following three steps:

FAST keypoint detection
FAST is very efficient approach for finding key-points, but it has several disadvantages. In [14] Rublee et al. found that FAST is sensitive along edges. Therefore they implement FAST with lower threshold value to get enough corners (key-points), then filter them with Harris corner measures [29] to select strong keypoints. To obtain scale stable corners, they implement FAST on each level of scale pyramid of the input image.

FAST keypoint orientation
The FAST detector provides the locations of corner points, but it does not give any information about orientation of corners. To be rotation-invariant, the main orientation should be determined. To do that, Rosin's corner intensity [67] is employed.

Descriptor Generation
To describe the key-point, the ORB uses the BRIEF descriptor. While the BRIEF descriptor is not rotation invariant. To be rotation invariant, ORB integrate rotation invariance to the

BRIEF descriptor. The BRIEF descriptor choose pair pixel points according to predefined sampling pattern in the patch of interesting points. To each pair pixels, an intensity comparison is implemented. If the intensity of the first pixel of the pair bigger than the intensity of second one, 1 is assigned to the result, otherwise 0. After applying this to all chosen pairs, a binary descriptor will be generated by concatenating all results.

3.6 Retrieval strategies, Feature fusion and indexing

There exist two important factors which affect the quality and efficiency of retrieved results. The first one is the retrieval ability of the descriptors. The other one is the specific implementation of these methods. In the following part, we will talk about feature fusion methods, strategies for retrieval and feature indexing methods.

3.6.1 Retrieval strategies

To develop a well-performing, and time and memory efficient TR system, studies have presented various retrieval strategies such as off-line and on-line structure, phase-based methods and feature fusions. In the following part, we introduce them.

Off-line and on-line structure
To be runtime time and memory efficient, a TR system for large scale trademark retrieval is structurally composed of an *off-line* and an *on-line* part. To ensure efficiency and effectiveness, in the off-line part, certain procedures such as image processing, feature extraction, indexing, etc. are applied on millions of images. In contrast, in the on-line part, these procedures are just implemented to the query image. This enables us to get rid of high CPU computation and memory allocation problems.

Phase-based method
The phase-based method applies low-cost methods in the first phase for filtering out the non-candidate similar objects, and runs the high-cost method in the second phase to improve the quality of the retrieved results. Wang et al. [84] apply global features in the first phase and local features in the second phase, where they select Zernike moments as global and SIFT as local descriptor. Similarly Jain et al. [36] apply traditional text-based retrieval systems and a weighted combination of low level features in the first phase and the high cost deformable template matching process in the second phase.

3.6.2 Feature-level and decision-level fusion

Feature fusion is another popular technique in TR systems. Both low level and high level features have their own merits and fallacies. Naturally, all of them are prone to failure under certain conditions. Feature fusion is an effective way to overcome this challenge, and have been applied in related works such as [24, 64, 70, 86, 90]. Fusion techniques are divided into two groups in [42]: *uni-modal* and *multi-modal*. The most important difference between these two models is how they apply feature fusion. The uni-modal fusion is a feature oriented method, whereas the multi-modal approach is a retrieval result oriented method.

In uni-modal fusion, features from different domains and modalities are combined in a single vector. This makes the retrieving process more efficient, but the feature combination more difficult due to the necessity of meaningful normalization. Compared to the former, multimodal is better suited to dealing with diverse information. The advantage of this model is considering the retrieval strengths of the features in combination. The combination is implemented in two distinct ways. One of them combines similarity distance, the other one combines the ranking results. In our experiments, we implemented linear regression algorithm, Gradient descent, method from first group, and Inverse Rank Position (IRP), Borda Count (BC) and Leave Out (LO) algorithms from second group. We will introduce these methods in the following part.

Gradient descent
Gradient descent [9] is a linear regression method designed to find the local maximum or minimum ($F(x) = 0$) of function $f(x)$. In practice, it is unpractical to solve $F(x) = 0$ directly because of the high dimensionality. To solve that, an iterative searching is implemented in gradient descent.

In our experiments, we apply gradient descent for finding the optimal weights for feature fusion. Feature fusion method combines similarity distance matrix of query images with dataset images with given weight values. Our similarity distance matrix is a 320×930328 matrix, which is quite large. Therefore, to implement Gradient descent, we approximate it as follows:

- First, we set all n weights $w_1, w_2, ..., w_n$ to 1 (to speed up conversion heuristically chosen values can also be used) and calculate a new similarity distance matrix D of p query images with q dataset images:

$$D^j = \sum_{i=1}^{n}(w_i^j * D_i), \tag{3.18}$$

  where D_i is similarity distance matrix of feature i, and D^j is combined similarity distance matrix and w_i^j is the weight value of feature i at iteration j.

- In the second step, we have to update each weight value in order. The new weight value w_i^{j+1} is decided by the step size γ and the change at value of function f when current weight value w_i^j is replaced by its neighborhood weight $\widetilde{w}_i^j$.

$$\widetilde{w}_i^j = w_i^j * (1 + \varepsilon), \tag{3.19}$$

$$w_i^{j+1} = w_i^j - \gamma * \Delta, \tag{3.20}$$

$$\Delta = f_{i+1}^j - f_i^j, \tag{3.21}$$

  where f_i^j is calculated by using the recently updated weights w_1^{j+1} to w_i^{j+1} (updating of weights is asynchronous) and non-updated weights w_i^j to w_n^j.

The first iteration will be finished after updating all the weight values. In the following iteration, we repeat the second step until the difference of rank results of two successive iterations, $\mathit{\Delta}$, is under a threshold.

$$\mathit{\Delta} = f_n^{j+1} - f_n^j \tag{3.22}$$

The gradient descent method finds the best ranking results by using the similarity distance matrix calculated according to the found optimal weight values. In contrast, Inverse rank position (IRP), Borda count (BC) and Leave out (LO) algorithms compared in [39] fuse rank results (i.e., decisions) calculated from different similarity distance matrix. In the following part, we briefly introduce these algorithms.

Inverse rank position (IRP)
This algorithm merges the ranking list of multiple features by getting inverse of the sum of inverse of multi-feature similarity ranks of image i for query image q as:

$$IRP(q,i) = \frac{1}{\sum_j^n \frac{1}{rank_j}}, \tag{3.23}$$

where j represents the j^{th} feature.

This algorithm is robust in large scale multi-feature merging. Because the difference of inverse of numbers which are bigger than certain threshold are very small, therefore, the extreme outliers of retrieval results affect the quality of retrieval results only slightly. For instance, SIFT algorithm fails at detecting similarity of some images. In these cases, the ranking position of compared images are at the end of the ranking list. In some fusion methods, these results will reduce the good rankings of SIFT features.

Borda count (BC)
Borda count considers fusion rank results as sum of number of votes from fused features. The votes of feature j for image i for being similar to query q is the similarity ranking number of image i when compared to query q with employing the feature j:

$$IRP(q,i) = \sum_j^n rank_j, \tag{3.24}$$

where j represents the j^{th} feature.

Leave out (LO)
Leave out algorithm merges ranking lists of multi features into a single rank list. In this method, each feature will insert an image from its ranking list one by one to bottom of the single rank list. This image should be owns highest rank in the list and also be not in the single rank list. The insertion order is defined heuristically.

3.6.3 Indexing

Processing millions of feature vectors is invincible under ordinary platform. Indexing these features is a possible solution to overcome this bottleneck. The Bag of Visual Words model in combination with $tf - idf$ and inverted index is a well-developed model for large scale image retrieving, which is introduced in the following part.

BoVW
In large scale image datasets, the local feature descriptor extracts millions of local features. For instance, local feature descriptors extracted around 500 millions features from our dataset. Employing local feature vectors directly has two main disadvantages: 1. Millions of local

feature vectors require large memory and disk spaces (*e.g.*., the smallest feature vectors of our dataset in binary format takes around 10GB memory-space in total). 2. Matching local features one by one with a distance metric for finding nearest neighbors is impractical.

It is known that text search has achieved quite progress and it is possible to access the desired results in seconds or milliseconds from numerous text files. One of the key factors leading to the success in text search is the BoW (bag of words) model [16]. In this model, stems of meaningful words or phrases of text documents are represented by indices after mapping with a hashing function. Then local and global frequencies of each words are calculated and documents are represented with their indices term-frequency vector. To match two document files, term-frequency vector similarities are calculated by appropriate metric distance, such as Euclidean distance.

In [77] retrieval of images are recast as text retrieval. This method is referred as BoVW (bag of visual words), where each local key-point descriptor vector is indexed as a visual word and the collections of distinctive visual words are visual code book of the image dataset. To index each feature vector, the quantization procedure of feature vectors space is finished by classifying local key-point features with classifier such K-means classification [20].

The BoVW method overcomes the two shortcomings of traditional nearest neighbor method. It describes a feature vector with a single index by mapping according to visual code book. In that case, cumbersome feature datasets become compact. Another important advantage is shifting the costly on-line computation to off-line by pre-computing.

3.6.4 TF-IDF

In the BoVW model, each image is represented by a vector of visual word occurring frequencies. Without applying a proper weighting method, computing similarity of images with vectors of occurring frequencies may not return accurate results. The reason is that the importance of each visual word to the image is not equal, the term with high frequency may have low weighting factor if it has high frequency in all images of the dataset, while a term with low frequency with high weighting factor is also possible if it only appeared in few documents in all dataset. To assign weighting factors to the words, the tf-idf (term frequency-inverse document frequency) [16], a standard weighting method in information retrieval and text mining, is employed. The key idea behind this method is describing a file with important factors of words to the document. The important factor of the term i in document j is decided by two statistics, term frequency tf and inverse document frequency idf, which are defined as:

$$tf_i = \frac{\text{number of occurence of term i in document j}}{\text{number of terms in document j}}, \tag{3.25}$$

$$idf_i = \log \frac{\text{Number of documents include term i}}{\text{Number of documents in dataset}}, \tag{3.26}$$

where t_i, the tf-idf value of term i in document j is defined as:

$$t_i = tf_i * idf_i. \tag{3.27}$$

The $tf - idf$ feature vector of a document of N terms will be $< t_1, t_2, t_3, \ldots, t_n >$. In the retrieval stage, the $tf - idf$ feature vectors will be used to achieve similarity of documents. The similarity of $tf - idf$ feature vectors is calculated with Cosine vector distance in Appendix B in this work.

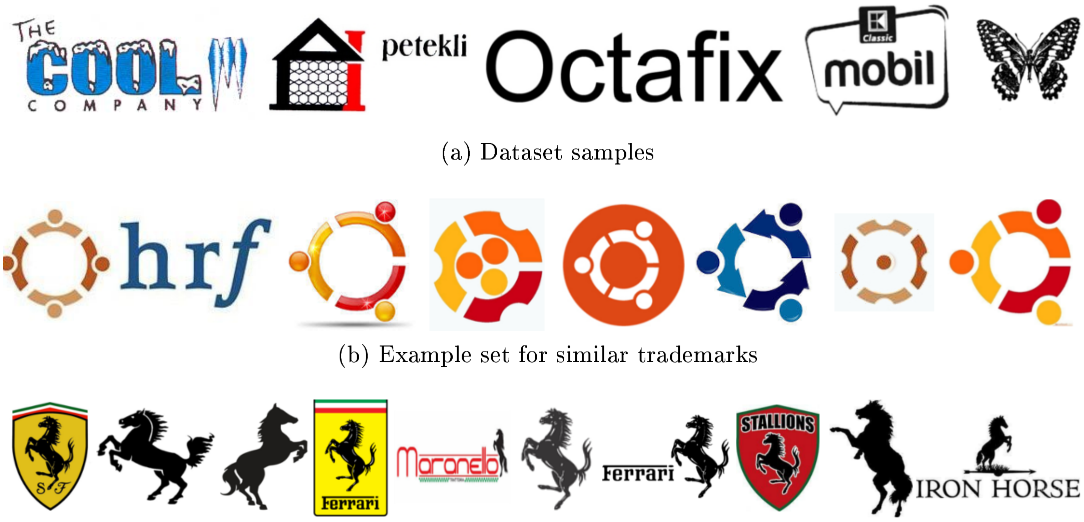
3.6.5 Inverted index

In the BoVW model, similarities of documents are calculated with $tf - idf$ feature vectors. There is a trade-off on time efficiency and memory efficiency when size of unique visual words K is large. If we save all $tf - idf$ feature vectors in the feature vector matrix, a very large sparse feature vector matrix will be generated. Because each document only owns N unique words, and N is very small compared to K. These sparse feature matrices cause a memory problem in TR systems. While calculating $tf - idf$ feature vectors each time is redundant and time consuming due to searching of whole BoVW index list to obtain statistical information for calculating $tf - idf$ value.

The inverted file structures are not only memory efficient, but also practical. Inverted file structure is developed in [77] where inverted file structure of each word will store the occurrences of the word in all documents. As a consequence, this inverted structure has saved the time of calculating occurrence by searching features of dataset.

CHAPTER 4

THE METU TRADEMARK DATASET



(a) Dataset samples

(b) Example set for similar trademarks

(c) Another example set for similar trademarks

Figure 4.1: Samples from our dataset. (a) Arbitrary samples. (b) Example for similar trademarks. (c) Another example set for similar trademarks.

One of the main contributions in this thesis is a large-scale challenging trademark dataset [81, 82]. The dataset contains a subset for investigating color in trademark retrieval.

4.1 The Dataset

Our main trademark dataset includes 930,328 images, corresponding to 409,834 many different company trademarks. The dataset is provided by the patent office “Grup Ofis Marka Patent A.Ş." [1] and extended with our own examples. As illustrated in Figure 4.1, the dataset includes trademarks of different companies that not only include colored shapes but also text of various forms. The details of the dataset are described in Table 4.1.

From the dataset, we have selected 320 similar trademark sets (similar to those in Figures 4.1b and 4.1c) as having “known” similarities. We will use images in these sets as queries and expect to find the labeled similar images among the 930,328 images. We see that it is far

[1] http://www.grupofis.com.tr

from trivial to find similarities by using local features only and that a good approach should combine color, texture, text, shape and parts information as much as possible, in addition to the global information, to be able to achieve good performance.

Table 4.1: Details of our dataset.

Aspect	**Value**
# trademarks	930,328
# unique registered firms	409,834
# unique trademarks	691,149
# trademarks containing text only	589,562
# trademarks containing figure only	19,394
# trademarks containing figure and text	312,154
# other trademarks	8,942
# file format	JPEG
# Max Resolution	1800 × 1800(px)
# Min Resolution	30 × 30(px)

4.1.1 The color subset

As mentioned above, our query dataset involves various similarity factors such as shape, texture, color etc. Therefore, it is not appropriate to evaluate color methods on the main dataset. For this reason, we set up a small scale color dataset including 600 trademarks in 10 different colors; red, green, blue, cyan, yellow, pink, black, gray, orange and brown. See Figure 4.2 for the samples.

4.2 Comparison to other datasets

As we mentioned in related works at Section 2.2.2, there have been many studies on automated trademark retrieval. However, the datasets used by these studies are very trivial and far from being a challenge for the feature descriptors that have proven useful in Computer Vision and Pattern Recognition. This is shown in Table 4.2, where the size and the type of datasets are listed in detail.

4.3 Challenges in the dataset

Our dataset own several challenges: 1. The number of trademarks is very big, it is almost close to 1 million. Trademark retrieval algorithms have to sacrifice their retrieval quality for efficiency in some range. Therefore achieving outstanding retrieval results in our dataset is very difficult. 2. The query sets contain various similarity aspects and close real life cases. 3. The images of datasets contain noises and is not auto-cropped. In our experiment, we developed our own enhanced auto-crop methods, but there still exist non fully auto-cropped images because of high noises. 4. Dataset includes other trademark which are similar to trademarks from query sets.

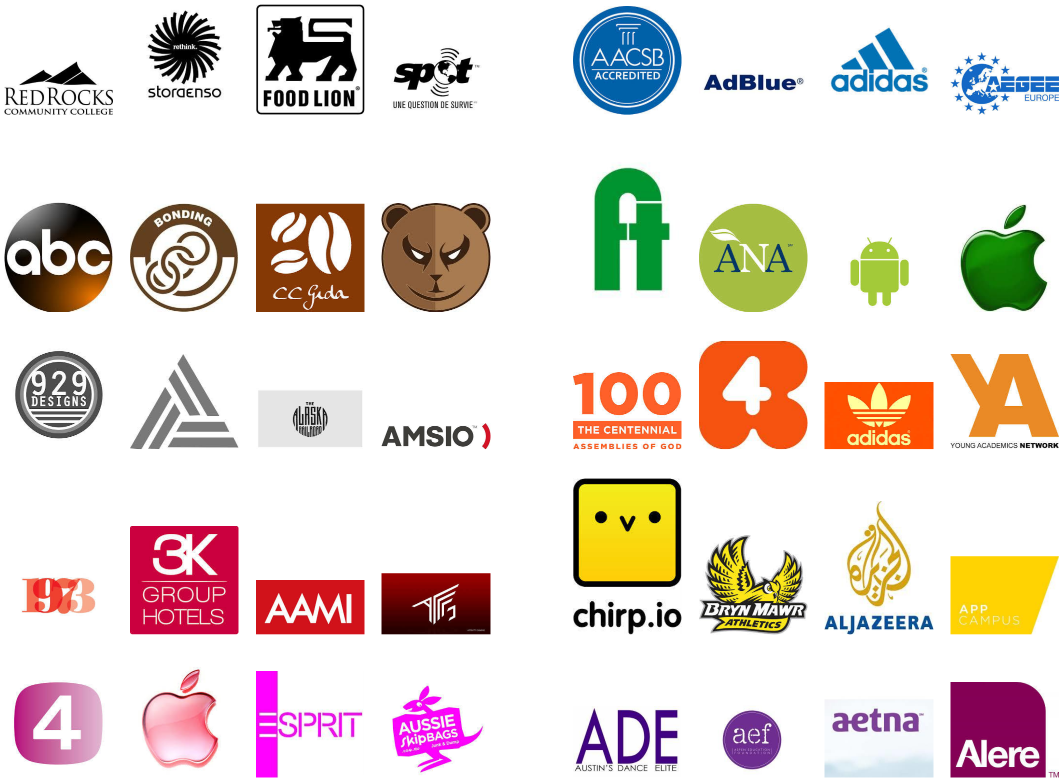


Figure 4.2: Trademark samples from our color dataset having different colors. Each color set includes four trademark samples, and these color sets are black, blue, brown, green, gray, orange, red, yellow, pink, purple from left to right up to down respectively.

Table 4.2: A comparison of trademark datasets available in the literature.

Dataset	Number of Logos	Image Type	Image Size	Ref.
U. of Maryland	106	BW	various	[55]
MPEG7 CE Shape-2 Part- B	3,621	BW	-	[84]
Wei et al.	1,003	BW	200 × 200(px)	[86]
Alwis^1 et al.	210	BW		[5]
Alwis^2 et al.	1000	BW	-	[7]
US Patent and Trademark office dataset	63,718	BW	-	[2]
MPEG7 CE Shape- 1 Part-B	1,400	BW	256×256	[63]
MPEG7	3,000	BW	-	[37]
Her et al.	2,020	RGB	64 × 64	[30]
Jain et al.	1,100	BW	200 × 200 500 × 500	[17, 36]
UK Trademark Registry	10,745	BW	-	[21]
Leung et al.	2,000	BW	-	[47]
METU Dataset	**930,328**	**RGB**	**various**	**[82]**

CHAPTER 5

EXPERIMENTS AND RESULTS

Another main contribution of the thesis is evaluating retrieval performance of various image descriptors on our large scale METU dataset. In this section, we introduce the experiment platform and configurations of settings. In addition, we demonstrate the precision-recall, ranking, performance time, and visual results of our experiments and analyze these results.

5.1 Experiment setup

The experiments were performed on a machine with an Intel i7-4770K 3.50GHz model CPU and 16GB DDR3 memory. Most experiments are done on MATLAB, while time and memory consuming processes such as large-scale K-means clustering and extraction of some features such as ORB were implemented in C/C++ using OpenCV.

5.2 Evaluation

To evaluate the results, we selected precision-recall (PR), *average rank*, *Rank*, and *normalized average rank*, $\widetilde{Rank}$ as evaluation measures:

$$Precision = \frac{No.\ of\ relevant\ retrieved\ Trademarks}{No.\ of\ retrieved\ Trademarks}. \tag{5.1}$$

$$Recall = \frac{No.\ of\ relevant\ retrieved\ Trademarks}{No.of\ relevant\ Trademarks}. \tag{5.2}$$

The precision-recall measure is a popular way of showing the quality of top retrieved results. The retrieval results are considered as outstanding when the precision value is close to 1 even if the recall value is increasing. It assigns more credits to the top retrieved results and fewer credits to the outliers. In other words, it is less affected by the outliers of retrieved results.

Compared to the precision-recall evaluation method, the ranking measures emphasize the performance of overall results. Therefore, it is more affected by the outliers. To minimize this shortcoming, we presented standard deviations of the ranking results and graphs of individual ranking results. Average rank of a retrieval is defined as follows:

$$Rank = \frac{1}{N_{rel}} \sum_{i=1}^{N_{rel}} R_i, \tag{5.3}$$

where N_{rel} is the number of relevant images for particular query image, N is the size of the image set, and R_i is the rank of the i^{th} relevant image.

The normalized average rank [53] is another evaluation method for analyzing the performance of retrieval systems:

$$\widetilde{Rank} = \frac{1}{N \times N_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2} \right). \qquad (5.4)$$

Average rank measure takes values in the range from $1 + \frac{N_{rel}}{2}$ to $N - \frac{N_{rel}}{2}$, where the smallest rank corresponds to the best retrieval. In contrast, the normalized rank measure lies in the range $[0, 1]$. Zero (0) means the best retrieval performance, and 0.5 corresponds to random retrieval, and one (1) is the worst performance.

5.3 Effect of the Parameters

The retrieval performance of each implemented method is influenced by its parameters. To achieve best performance, we tested some of the methods with different parameters. In the following part, we show these results.

5.3.1 Color Histogram Parameters

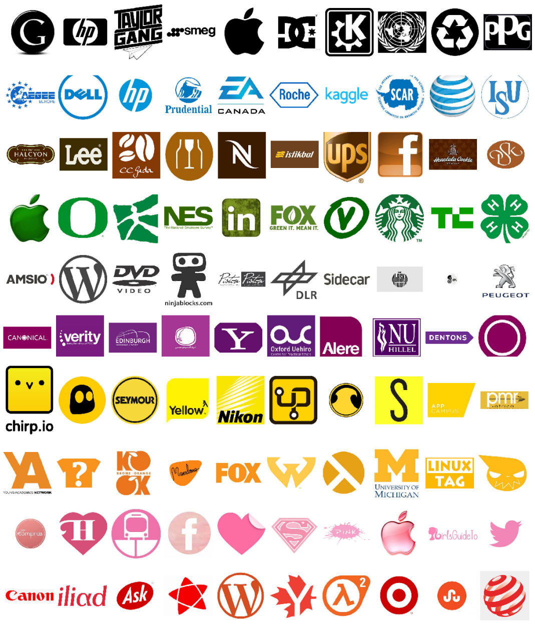


Figure 5.1: Retrieval results of the color histogram method on the color dataset (Note: the first image of each row is the query image and other images are retrieved images).

As we mentioned in Chapter 3, the color histogram method is affected by three factors: color space, normalization and distance metrics. To find out the best setting for the color histogram method, we tested RGB, HSV color spaces, $L1$ and $L2$ normalization methods, and 5 different distance metrics (Manhattan, Euclidean, Intersection, Cosine vector, and Quadratic distance).

Moreover, quantization of the color model is needed. We quantize the RGB color model into 64 and 512 bins by dividing R, G, B channels into 4 and 8 uniform parts. In contrast, for the HSV color model, the non-uniform quantization described in [46] and Li et al. [28] is applied.

Here we present various combination of the aforementioned factors on our color dataset. Figure 5.2e shows the precision-recall results of 64- and 512-bin RGB color histogram, 36- and 72-bin HSV color histogram and the best among them. We see that the 36 bins with $L1$ normalization and intersection distance shows the best performance. In spite of using the smallest color histogram vector, the 36-bin HSV color histogram method gives outstanding performance. This is a memory and time efficient method. See Figure 5.1 for sample retrieval results of the best performing color histogram method on our color dataset.

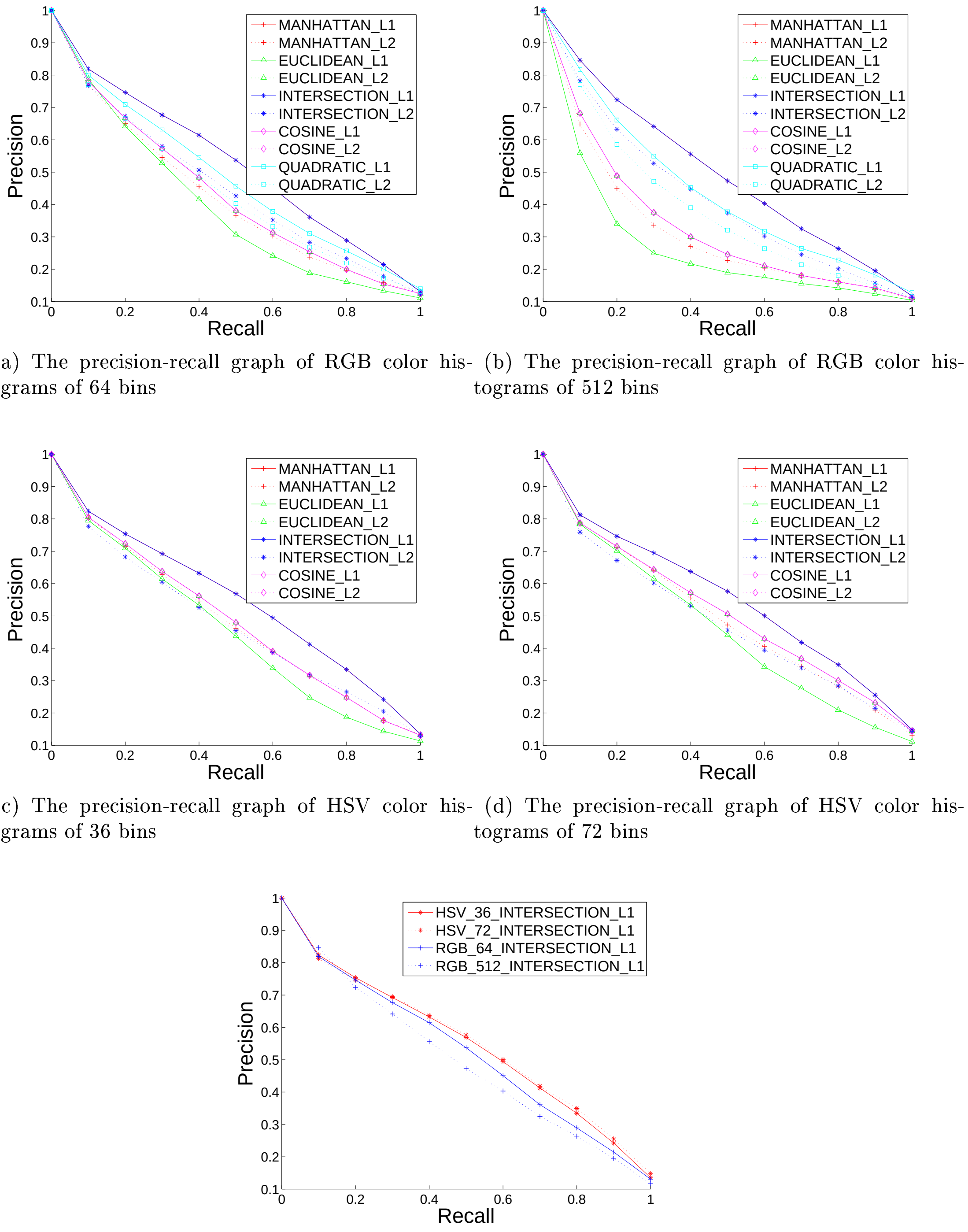
To test retrieval ability of the color histogram method, we implemented color histogram method with the best parameters on the METU dataset. When we only consider color similarity, color histogram method performed very well. While if we take the ranking and precision-recall results in Table 5.1 and Figure 5.4 into our evaluation, the performance is far from desired. That is because most of our query images own diverse color scheme. However, this does not conclude that the color histogram method is not useful. Figure 5.5 shows that certain individual results of color histogram method are well-performed. That is because query images of that set have close color schemes. In summary, the color histogram method performs very well if the similarity of trademarks relies on color. Therefore, applying color schemes adaptively (based on the color content of the trademarks) should be preferred.

5.3.2 LBP Parameters

Four different LBP methods, $LBP_{P,r}$, $LBP^{ri}_{P,r}$, $LBP^{u2}_{P,r}$, $LBP^{riu2}_{P,r}$, were introduced in Section 3.4. We have implemented these operators with different normalization and distance metrics. We tested $L1$ and $L2$ normalization methods, and 4 different distance metrics (Manhattan, Euclidean, Intersection, Cosine vector distance).

Figure 5.3 shows precision-recall curves of $LBP_{P,r}$, $LBP^{ri}_{P,r}$, $LBP^{u2}_{P,r}$, $LBP^{riu2}_{P,r}$ with different normalizations and distance metrics. Among them, the $LBP_{P,r}$ with $L1$ normalization with cosine metric distance shows the best result. Moreover, the $LBP^{u2}_{P,r}$ shows good performance. This is because these two methods own adequate distinctive patterns compared to others and trademark infringement by rotation is not very common in trademark similarity.

Figure 5.4 shows the precision-recall performance of the best performing LBP pattern method being close to local features such as HOG and ORB. Compared to them, LBP is time and memory efficient both in off-line and on-line processing. To further improve the retrieval quality of LBP, we could apply fusion methods on these four types of LBP methods, since all of them are time and memory efficient. In that case, the LBP method would not only own enough distinctiveness but also gain rotation invariance.



(a) The precision-recall graph of RGB color histograms of 64 bins

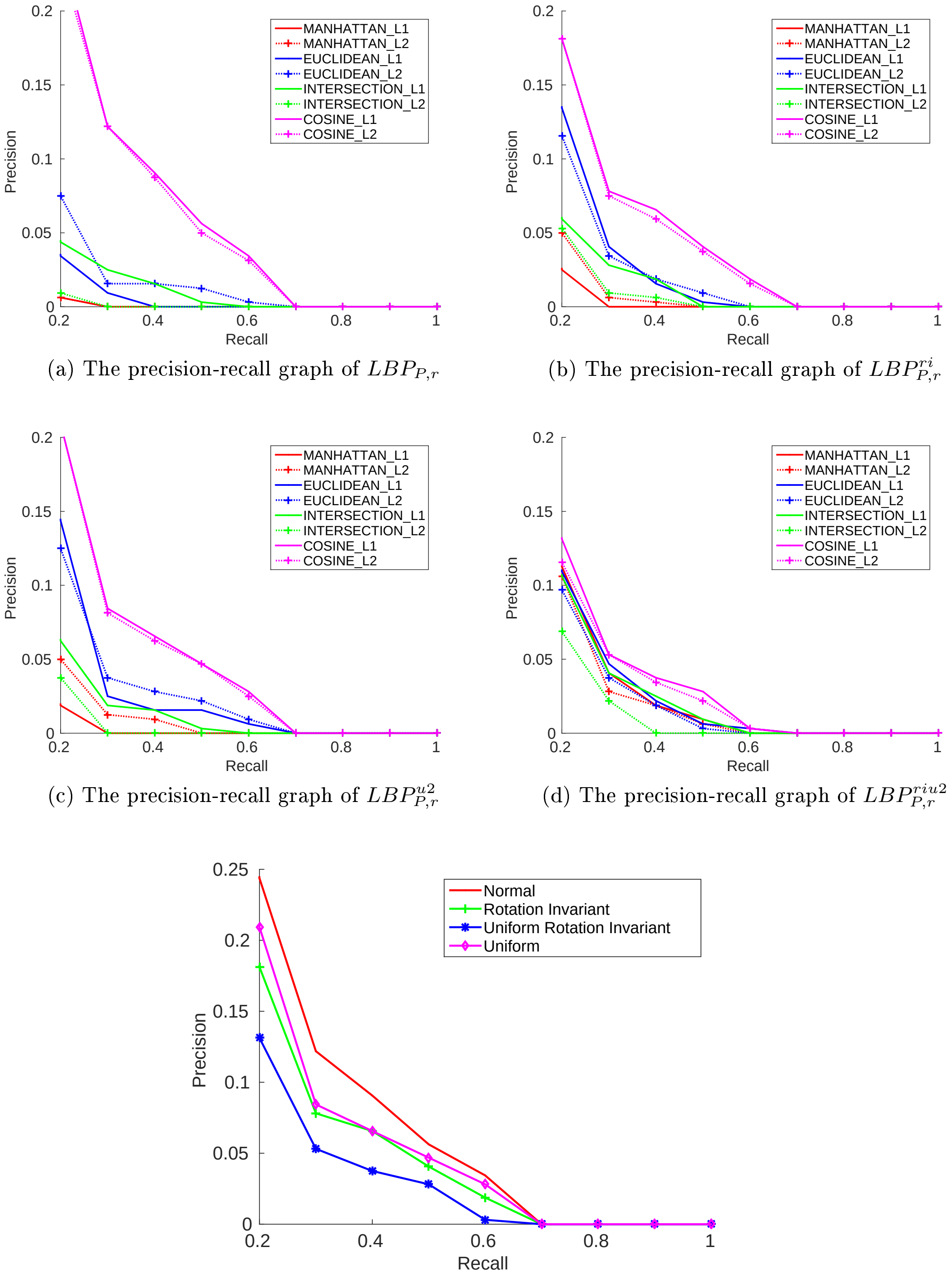
(b) The precision-recall graph of RGB color histograms of 512 bins

(c) The precision-recall graph of HSV color histograms of 36 bins

(d) The precision-recall graph of HSV color histograms of 72 bins

(e) The comparison of outstanding schemes from (a-d)

Figure 5.2: The precision-recall results for trademark retrieval in the 600-trademark dataset, grouped by the utilized normalization scheme and color space. (a-d) The results of RGB color histograms of 64 and 512 bins and HSV color histograms of 36 and 72 bins, compared for various distance metrics and normalization. (e) A comparison of the best overall results. The numeric prefixes in the legend entries denote the number of quantization bins, while the string suffixes indicate the utilized distance metric and normalization.



(a) The precision-recall graph of $LBP_{P,r}$

(b) The precision-recall graph of $LBP^{ri}_{P,r}$

(c) The precision-recall graph of $LBP^{u2}_{P,r}$

(d) The precision-recall graph of $LBP^{riu2}_{P,r}$

(e) The comparison of outstanding schemes from (a-d)

Figure 5.3: The precision-recall results of four LBP variants on the METU dataset. (a-d) The results of $LBP_{P,r}$, $LBP^{ri}_{P,r}$, $LBP^{u2}_{P,r}$, $LBP^{riu2}_{P,r}$. (e) A comparison of the best overall results. In legends of (a-d), the string suffixes indicate the utilized distance metric and normalization type.

5.4 Retrieval Results of Individual Features

We implemented global descriptors such as color histogram, edge orientation histogram, LBP and Shapemes and local feature point descriptors such as SIFT, SURF, Triangular SIFT, ORB and HOG. The implementations of Shapemes and local feature descriptors are integrated with the BoVW method. To build visual vocabulary book for the BoVW method, histogram vector of Shapemes and local feature key-points are quantized by K-means clustering. Most of these methods extract around 500 million features from trademarks in the METU dataset. Classifying datasets of this scale is challenging in terms of CPU time and memory. To be time and memory efficient and to have enough distinctive visual vocabulary book, the descriptors are clustered into 10,000 classes. However, we clustered Triangular SIFT into 1,000 classes, because 1,000 different local features would generate 1,000,000,000 different triplet groups. Since our computational facilities were not sufficient, for Triangular SIFT, we only used 1,000 classes. For the sake of fairness, we also implemented SIFT method with 1,000 visual words.

Ranking and precision-recall results of all implemented methods are compared in Table 5.1 and Figure 5.4. To overcome the limitation of average ranking $Rank$ and normalized average rank $\widetilde{Rank}$, we displayed $\widetilde{Rank}$ of 320 query images in Figure 5.5. In this figure, points close to X axis mean better retrieval results.

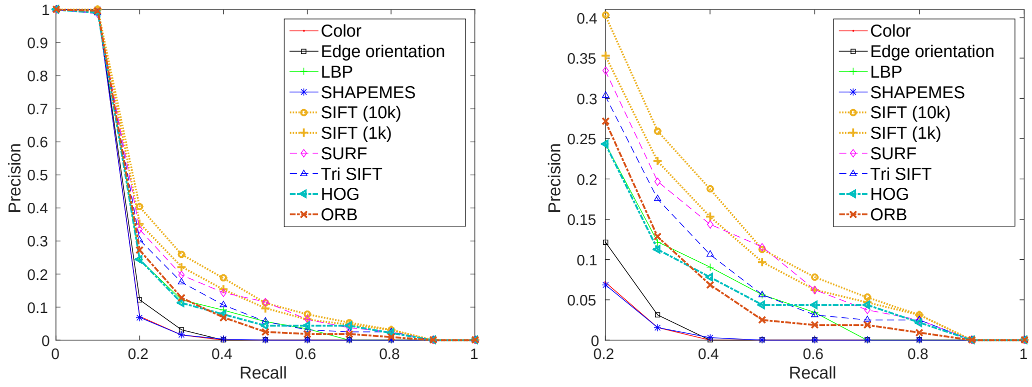
According to the results, we conclude that local descriptors such as SIFT, Triangular SIFT and HOG are better than global features such as color, edge orientation and LBP in rank and precision-recall values. Among the local features, Triangular SIFT has the best ranking results and SIFT has the best precision-recall result. Upon a closer look in Figure 5.5, we see that many SIFT points are closer to X axis than Triangular SIFT and other methods. This, together with the precision-recall values, suggests that the retrieval quality of SIFT is better than other algorithms.

Table 5.1: Comparison of the individual methods on the METU trademark dataset.

Algorithm	**Average rank**	**Normalized**
(BoVW dictionary size)		**average rank**
Color	314,953.2 ± 194,291.3	0.339 ± 0.209
Edge orientation	350,662.1 ± 149,797.4	0.377 ± 0.161
LBP	244,830.3 ± 133,210.8	0.263 ± 0.143
SHAPEMES (10k)	141,489.5 ± 117,323.3	0.152 ± 0.126
HOG (10k)	142,337.5 ± 91,221.6	0.153 ± 0.098
SIFT (10k)	141,994.2 ± 116,035.8	0.153 ± 0.125
SIFT (1k)	182,343.3 ± 149,526.3	0.196 ± 0.161
SURF (10k)	107,700.1 ± 95,073.8	0.116 ± 0.102
TRI-SIFT (1k)	**66,117.9 ± 64,736.4**	**0.071 ± 0.070**
ORB (10k)	130,043.3 ± 88,567.9	0.140 ± 0.095

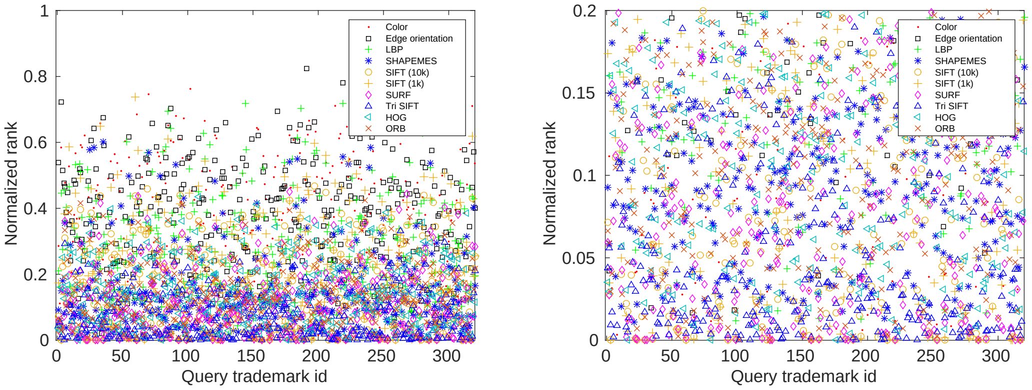
5.5 Evaluation of Running times

The performance of running times of TR systems includes off-line and on-line processing times. The off-line process includes feature extraction and clustering (due to BoVW). Table 5.3 shows the feature extraction times of methods tested in the thesis. Among them, ORB



(a) Original (b) Zoomed

Figure 5.4: Precision-recall results of methods (single) on the METU dataset.



(a) Original (b) Zoomed

Figure 5.5: Normalized average ranking results of methods (single) on the METU dataset.

feature has the best feature extraction time: ORB features from 1 million trademark images can be extracted around 10 minutes with parallel OpenCV implementation.

The on-line processing time is more critical for the TR systems. The on-line processing time is the sum of feature extraction time for the query image and the matching time. Table 5.2 shows the matching time of the implemented methods. Low-level features such as color histogram, edge orientation histogram and LBP take around 100 milliseconds. However, high-level features except for Triangular SIFT take around 10 seconds with a parallel Matlab implementation. The matching time is mainly effected by the matching method and the size of descriptors. See Table 5.4 for matching method and size of descriptors.

5.6 Results of fusion

To further improve retrieval results, four kind of fusion methods are tested. In all implementations, we fuse Triangular SIFT, SIFT, HOG, Shapemes and color histogram. In the Gradient descent implementation, initially, we assign the same weight to these five methods. After 100

Table 5.2: Running time of the methods on our dataset. The table lists only the matching time of a query to 930,328 trademarks in the dataset (in Matlab).

Algorithm (BoVW dictionary size)	**Time (milliseconds)**	**Parallel process**
Color	141.6	-
Edge orientation	100.3	-
LBP	49.4	-
HOG (10k)	16,036.4	Yes
SHAPEMES (10k)	18,567.7	Yes
SIFT (10k)	19,195.9	Yes
SIFT (1k)	7,659.9	Yes
SURF (10k)	15,639.1	Yes
ORB (10k)	10,591.2	Yes
TRI-SIFT (1k)	53,292.8	Yes

Table 5.3: Feature extraction time of the tested methods.

Algorithm	**Type/BoVW Dictionary Size**	**Tool**	**Average feature extraction time (seconds)**	**Standard deviation of feature extraction time (seconds)**	**total feature extraction time (hours)**
Color Histogram	RGB 64 bins	Matlab	0.0943	0.1379	24.37
Color Histogram	RGB 512 bins	Matlab	0.0944	0.1370	24.38
Color Histogram	HSV 36 bins	Matlab	0.0389	0.0646	10.05
Color Histogram	HSV 72 bins	Matlab	0.0364	0.0641	9.41
LBP	Rotation invariant	Matlab	0.0704	0.1108	18.18
LBP	Uniform rotation invariant	Matlab	0.0685	0.1077	17.70
LBP	Uniform	Matlab	0.0519	0.1009	13.41
LBP	Normal	Matlab	0.0309	0.0509	7.99
HOG	Fast 100	Matlab	0.0545	0.0512	14.07
SIFT	10,000	Matlab	0.2232	0.2947	57.72
Tri-SIFT	1,000	Matlab	0.2799	0.3794	72.3
SURF	10,000	Matlab	0.0440	0.0582	11.38
Shape context	10,000	Matlab	0.1197	0.1073	30.93
ORB	10,000	Opencv	**0.0006**	**0.0000**	**0.17**

iterations, the weights of Shapemes, HOG, color histogram and Triangular SIFT are: 0.2777, 0.2822, 0, 0.2644, 0.1756.

Inverse Position and Borda Count method As we mentioned in Section 3.6, the Leave Out algorithm select an order for fusion. The order determined by our experiments is Triangular SIFT, HOG, Shapemes, SIFT, and color histogram, which is an random order. In the future, we will do more detailed experiments for finding a rule for making the order, which further improve the result of Leave our algorithm.

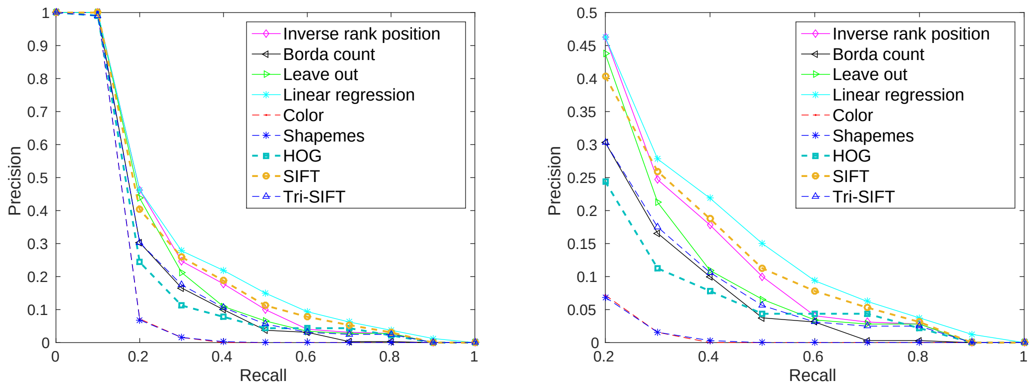
The ranking and precision-recall results of fusion are given in Table 5.5 and Figure 5.6. We see that the IRP algorithm shows best ranking results and the linear regression (Gradient descent) algorithm shows best precision-recall results. Besides that, See Figure 5.7 for checking the effect of fusion methods on individual ranking results.

Table 5.4: Size of the descriptors and the applied distance metrics.

Algorithm	**Type**	**Descriptor vector size**	**Maximum number of descriptors**	**Distance metric**
Color Histogram	RGB 64 bins	64	1	Intersection
Color Histogram	RGB 512 bins	512	1	Intersection
Color Histogram	HSV 36 bins	36	1	Intersection
Color Histogram	HSV 72 bins	72	1	Intersection
LBP	Rotation invariant	36	1	Cosine vector
LBP	Uniform rotation invariant	10	1	Cosine vector
LBP	Uniform	59	1	Cosine vector
LBP	Normal	256	1	Cosine vector
Edge orietaion	Normal	24	1	Cosine vector
Shapemes	Normal	60	1	Cosine vector
HOG	Normal	32	100	Cosine vector
ORB	Normal	256 (binary)	500	Cosine vector
SIFT	Normal	128	-	Cosine vector
SURF	Normal	64	-	Cosine vector

Table 5.5: Comparison of ranking results of feature-fusion methods on our dataset (Note: the first five features in the table are the to-be-fused features).

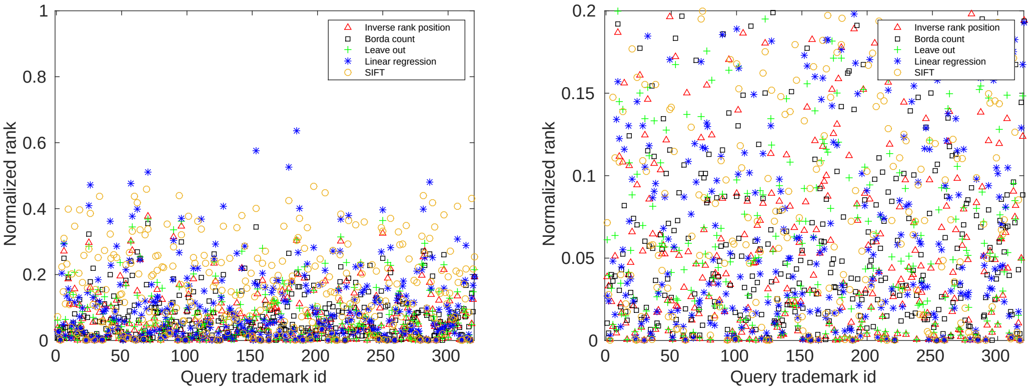
Algorithm (BoVW dictionary size)	**Average rank**	**Normalized Average rank**
Color	314,953.2 ± 194,291.3	0.339 ± 0.209
SHAPEMES (10k)	141,489.5 ± 117,323.3	0.152 ± 0.126
HOG (10k)	142,337.5 ± 91,221.6	0.153 ± 0.098
SIFT (10k)	141,994.2 ± 116,035.8	0.153 ± 0.125
TRI-SIFT (1k)	**66,117.9 ± 64,736.4**	**0.071 ± 0.070**
Linear regression	98,618.6 ± 107,040.8	0.106 ± 0.115
IRP	**60,426.1 ± 68,347.2**	**0.065 ± 0.073**
BC	66,782.4 ± 65,235.7	0.070 ± 0.070
LO	67,412.0 ± 74101.7	0.072 ± 0.080



(a) Original

(b) Zoomed

Figure 5.6: Comparison of Precision-recall results of fusion and single methods on the METU dataset.



(a) Original

(b) Zoomed

Figure 5.7: Normalized ranking results of fusion methods on the METU dataset.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, retrieving similar trademarks in a large trademark dataset is investigated using widely used methods in Computer Vision and Pattern Recognition. Automated tools for trademark retrieval have become very essential since, due to tremendous increase in trademark applications and registrations, traditional trademark query systems using manual methods have become insufficient, burdensome, inefficient and expensive. Owing to the absence of automated tools, trademark infringements and piracies have also increased in these decades.

The first contribution of the thesis was the introduction of a publicly-available large trademark dataset (METU Trademark Dataset) into the literature. The METU Trademark Dataset contains 930,328 images corresponding to 409,834 different company trademarks and sets a very challenging benchmark for not only trademark retrieval but also image retrieval studies. The dataset is crucial since currently available datasets are very small and far from being a challenge to the developed methodologies in Computer Vision and Pattern Recognition.

Moreover, the thesis compared widely-used methods in Computer Vision and Pattern Recognition for trademark retrieval. Namely, the thesis evaluates the performance of shape and texture descriptors like SIFT, SURF, ORB, HOG as well as more global descriptors like color histogram, edge orientation histogram, LBP and shape context. This contribution serves as a baseline for further studies that will be performed on the dataset. The thesis showed that the dataset is very challenging and at best, the methods can retrieve similarities only around 60,000 rank, on the average. Although this is faster and more efficient than scanning through all the dataset manually, the results show that there is still much space for progress in trademark retrieval.

The best performing method in the dataset has turned out to be an extension of SIFT and it utilizes 3-ary relations between SIFT features at the same scale. This allows the method to not only prune the search space by combining and comparing features at the same scale but also include spatial relationship between features. This suggests that local information combined together using spatial relationships in a robust manner might be a good choice for trademark retrieval.

6.1 FUTURE WORK

Although there have been promising attempts for automated trademark retrieval, the existing systems are far from being useful by an agency overseeing trademark infringements. It has turned out that searching for similarities between trademarks is very challenging, as also noted by [42, 73, 86], and the main reason for this is that similarity between trademarks might occur at very different levels, from local visual information like edges, corners, texture to their local combination to their global organization, exploiting processes such as Gestalt grouping used by human visual system. As a future work, it is important to try to define similarities that might occur at different levels and develop a unified single system that can address the challenges at these challenges.

REFERENCES

[1] Color theory. http://learn.colorotate.org/color-models/#.VNy5fDV9z7A. Accessed: 2015.02.12.

[2] M. Abdel-Mottaleb and R. Desai. Fast image retrieval using multi-scale edge representation of images, June 6 2000. US Patent 6,072,904.

[3] N. Alajlan. Retrieval of hand-sketched envelopes in logo images. In *Image Analysis and Recognition*, pages 436–446. Springer, 2007.

[4] N. Alajlan, M. S. Kamel, and G. Freeman. Multi-object image retrieval based on shape and topology. *Signal Processing: Image Communication*, 21(10):904–918, 2006.

[5] S. Alwis and J. Austin. A novel architecture for trademark image retrieval systems. In *Electronic Workshops in Computing*, page 285, 1998.

[6] S. Alwis and J. Austin. Trademark image retrieval using multiple features. *CIR-99: the challenge of image retrieval, Newcastle-upon-Tyne, UK*, 1999.

[7] T. Alwis. *Content-based retrieval of trademark images.* PhD thesis, University of York, 2000.

[8] F. M. Anuar, R. Setchi, and Y.-K. Lai. A conceptual model of trademark retrieval based on conceptual similarity. *Procedia Computer Science*, 22:450–459, 2013.

[9] M. Avriel. *Nonlinear programming: analysis and methods.* Courier Corporation, 2003.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[11] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, volume 2, page 3, 2000.

[12] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[13] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010.

[15] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[16] G. Chowdhury. *Introduction to modern information retrieval.* Facet publishing, 2010.

[17] G. Ciocca and R. Schettini. Similarity retrieval of trademark images. In *International Conference on Image Analysis and Processing, 1999. Proceedings.*, pages 915–920. IEEE, 1999.

[18] G. Ciocca and R. Schettini. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recognition*, 34(8):1639–1655, 2001.

[19] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[20] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* John Wiley & Sons, 2012.

[21] J. P. Eakins, J. M. Boardman, and M. E. Graham. Similarity retrieval of trademark images. *IEEE multimedia*, 5(2):53–63, 1998.

[22] J. P. Eakins, J. D. Edwards, K. J. Riley, and P. L. Rosin. Comparison of the effectiveness of alternative feature sets in shape retrieval of multicomponent images. In *Photonics West 2001-Electronic Imaging*, pages 196–207. International Society for Optics and Photonics, 2001.

[23] J. P. Eakins, M. E. Graham, J. M. Boardman, et al. *Retrieval of trademark images by shape feature.* British Library Research and Innovation Report 26, 1996.

[24] J. P. Eakins, K. J. Riley, and J. D. Edwards. Shape feature matching for trademark image retrieval. In *Image and Video Retrieval*, pages 28–38. Springer, 2003.

[25] J. P. Eakins, K. Shields, and J. Boardman. Artisan: a shape retrieval system based on boundary family indexing. In *Electronic Imaging: Science & Technology*, pages 17–28. International Society for Optics and Photonics, 1996.

[26] W. H. Equitz and W. Niblack. *Retrieving images from a database using texture-algorithms from the QBIC system.* IBM Research Division, 1994.

[27] N. es. Delaunay voronoi. `http://commons.wikimedia.org/wiki/File:Delaunay_Voronoi.png`. Accessed: 2015.02.12.

[28] L. Guohui, L. Wei, and C. Lihua. An image retrieval method based on color perceived feature. *Journal of Image and Graphics*, 3, 1999.

[29] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[30] I. Her, K. Mostafa, and H.-K. Hung. A hybrid trademark retrieval system using four-gray-level zernike moments and image compactness indices. *International Journal of Image Processing (IJIP)*, 4(6):631, 2011.

[31] S. Hsieh and K.-C. Fan. Multiple classifiers for color flag and trademark image retrieval. *IEEE Transactions on Image Processing*, 10(6):938–950, 2001.

[32] T. Huang, S. Mehrotra, and K. Ramchandran. Multimedia analysis and retrieval system (mars) project. 1997.

[33] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[34] T. P. Institute. Tpi statistics database. `http://www.tpe.gov.tr/TurkPatentEnstitusu/statistics/`, 2014. Accessed: 2015.01.14.

[35] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern recognition*, 29(8):1233–1244, 1996.

[36] A. K. Jain and A. Vailaya. Shape-based retrieval: A case study with trademark image databases. *Pattern recognition*, 31(9):1369–1390, 1998.

[37] H. Jiang, C.-W. Ngo, and H.-K. Tan. Gestalt-based feature similarity measure in trademark database. *Pattern Recognition*, 39(5):988–1001, 2006.

[38] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *17th ACM international conference on Multimedia*, pages 581–584, 2009.

[39] M. Jović, Y. Hatakeyama, F. Dong, and K. Hirota. Image retrieval based on similarity score fusion from feature similarity ranking lists. In *Fuzzy Systems and Knowledge Discovery*, pages 461–470. Springer, 2006.

[40] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis. Scalable triangulation-based logo recognition. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 20. ACM, 2011.

[41] T. Kato. Database architecture for content-based image retrieval. In *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology*, pages 112–123. International Society for Optics and Photonics, 1992.

[42] A. Kesidis and D. Karatzas. Logo and trademark recognition. *Handbook of Document Image Processing and Recognition*, pages 591–646, 2014.

[43] Y.-S. Kim and W.-Y. Kim. Content-based trademark retrieval system using a visually salient feature. *Image and Vision Computing*, 16(12):931–939, 1998.

[44] P. Kochakornjarupong. *Trademark image retrieval by local features.* PhD thesis, University of Glasgow, 2011.

[45] C. Lam, J. Wu, and B. Mehtre. Star–a system for trademark archival and retrieval. *World Patent Information*, 18(4), 1996.

[46] Z. Lei, L. Fuzong, and Z. Bo. A cbir method based on color-spatial feature. In *TENCON 99. Proceedings of the IEEE Region 10 Conference*, volume 1, pages 166–169. IEEE, 1999.

[47] W. H. Leung and T. Chen. Trademark retrieval using contour-skeleton stroke classification. In *2002 IEEE International Conference on Multimedia and Expo, 2002. ICME'02. Proceedings.*, volume 2, pages 517–520. IEEE, 2002.

[48] S. Loncaric. A survey of shape analysis techniques. *Pattern recognition*, 31(8):983–1001, 1998.

[49] V. LoTempio. Walgreens sues wegmans over trademark. `https://www.lotempiolaw.com/2010/11/articles/trademarks/walgreens-sues-wegmans-over-trademark/`, 2014. Accessed: 2015.01.21.

[50] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[51] mathmunch. Circling, squaring, and triangulating. `http://mathmunch.org/2013/05/08/circling-squaring-and-triangulating/`. Accessed: 2015.02.12.

[52] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, volume 1, pages I–723. IEEE, 2001.

[53] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel-evaluation in image retrieval. In *Image and Video Retrieval*, pages 38–49. Springer, 2002.

[54] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. Qbic project: querying images by content, using color, texture, and shape. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 173–187. International Society for Optics and Photonics, 1993.

[55] U. of Maryland Logo Dataset. Laboratory for language and media processing (lamp), online: `http://lampsrv02.umiacs.umd.edu/projdb/project.php?id=47`, Last accessed: 18 December, 2014.

[56] T. Ojala, M. Pietikäinen, and T. Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *Advances in Pattern Recognition—ICAPR 2001*, pages 399–408. Springer, 2001.

[57] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[58] W. I. P. Organization. *International Classification of the Figurative Elements of Marks.* Number 502. WIPO, 1973.

[59] W. I. P. Organization. *International Classification of the Figurative Elements of Marks.* Number 502. WIPO, 1973.

[60] W. I. P. Organization. Wipo statistics database. `http://www.wipo.int/ipstats/en/`, 2014. Accessed: 2015.01.14.

[61] T. Pavlidis. A review of algorithms for shape analysis. *Computer graphics and image processing*, 7(2):243–258, 1978.

[62] R. Phan and D. Androutsos. Content-based retrieval of logo and trademarks in unconstrained color image databases using color edge gradient co-occurrence histograms. *Computer Vision and Image Understanding*, 114(1):66–84, 2010.

[63] H. Qi, K. Li, Y. Shen, and W. Qu. An effective solution for trademark image retrieval by combining shape description and feature matching. *Pattern Recognition*, 43(6):2017–2027, 2010.

[64] S. Ravela and R. Manmatha. Multi-modal retrieval of trademark images using global similarity. Technical report, DTIC Document, 2005.

[65] D. O. Reilly. Hsv coordinate system in a hexacone. `http://derek.dkit.ie/gui_programming/colour/colour.html`. Accessed: 2015.02.12.

[66] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol. Scalable logo recognition in real-world images. In *1st ACM International Conference on Multimedia Retrieval*, pages 25:1–25:8, 2011.

[67] P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999.

[68] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[69] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *2011 IEEE International Conference on Computer Vision (ICCV),*, pages 2564–2571. IEEE, 2011.

[70] M. Rusinol, D. Aldavert, D. Karatzas, R. Toledo, and J. Lladós. Interactive trademark image retrieval by fusing semantic and visual content. In *Advances in Information Retrieval*, pages 314–325. Springer, 2011.

[71] M. Rusinol, F. Noorbakhsh, D. Karatzas, E. Valveny, and J. Lladós. Perceptual image retrieval by adding color information to the shape context descriptor. In *2010 20th International Conference on Pattern Recognition (ICPR),*, pages 1594–1597. IEEE, 2010.

[72] H. Sahbi, L. Ballan, G. Serra, and A. Del Bimbo. Context-dependent logo matching and recognition. *IEEE Transactions on Image Processing*, 22(3):1018–1031, Mar. 2013.

[73] J. Schietse, J. P. Eakins, and R. C. Veltkamp. Practice and challenges in trademark image retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 518–524. ACM, 2007.

[74] H. Shan. Mcdonald's sues china trademark body over logo dispute. `http://www.china.org.cn/business/2011-10/17/content_23645862.htm`, 2014. Accessed: 2015.01.21.

[75] L. Shapiro and G. C. Stockman. Computer vision. 2001. *ed: Prentice Hall*, 2001.

[76] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.*, pages 1470–1477. IEEE, 2003.

[77] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.*, pages 1470–1477. IEEE, 2003.

[78] J. R. Smith and S.-F. Chang. Integrated spatial and feature image query. *Multimedia systems*, 7(2):129–140, 1999.

[79] A. Soffer and H. Samet. Using negative shape features for logo similarity matching. In *Fourteenth International Conference on Pattern Recognition, 1998. Proceedings.*, volume 1, pages 571–573. IEEE, 1998.

[80] E. Sowards. Colors values and css. `http://www.erinsowards.com/articles/2011/01/colors.php`. Accessed: 2015.02.12.

[81] O. Tursun and S. Kalkan. A benchmark and large dataset for trademark retrieval metu dataset. `http://kovan.ceng.metu.edu.tr/LogoDataset/`, 2014. Accessed: 2015.01.21.

[82] O. Tursun and S. Kalkan. A challenging big dataset for benchmarking trademark retrieval. In *The 14th IAPR Conference on Machine Vision Applications (MVA)*, 2015.

[83] A. Vedaldi. Scale invariant feature transform. `www.vlfeat.org/api/sift.html`. Accessed: 2015.01.14.

[84] Z. Wang and K. Hong. A novel approach for trademark image retrieval by combining global features and local features. *Journal of Computational Information Systems*, 8(4):1633–1640, 2012.

[85] Wapcaplet. Hsv color space as a conical object. `http://commons.wikimedia.org/wiki/File:HSV_cone.jpg`. Accessed: 2015.02.12.

[86] C.-H. Wei, Y. Li, W.-Y. Chau, and C.-T. Li. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition*, 42(3):386–394, 2009.

[87] M. Wertheimer. Laws of organization in perceptual forms. *A source book of Gestalt psychology*, pages 71–88, 1938.

[88] J.-K. Wu, C.-P. Lam, B. M. Mehtre, Y. J. Gao, and A. D. Narasimhalu. Content-based retrieval for trademark registration. *Multimedia Tools and Applications*, 3(3):245–267, 1996.

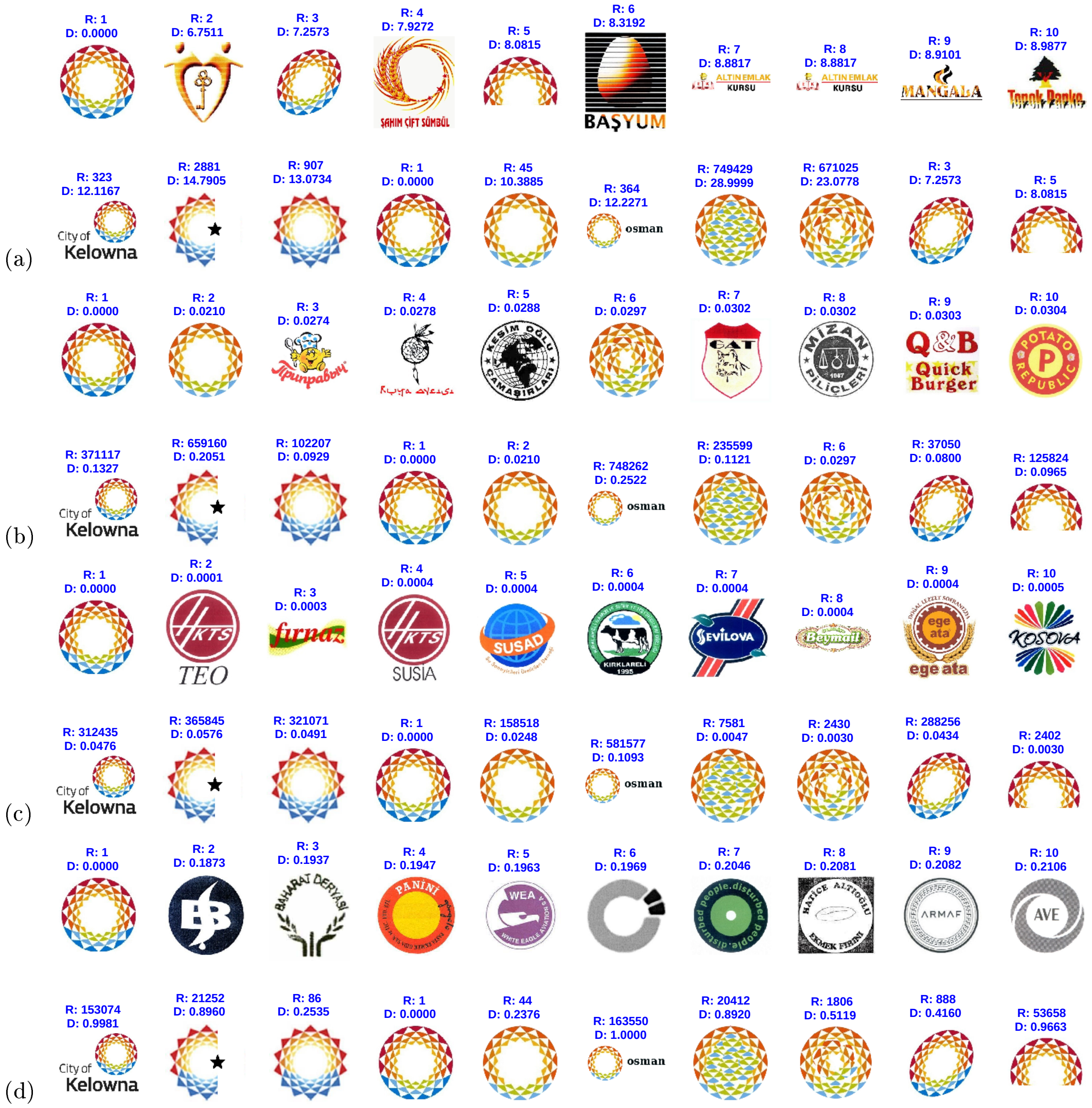
[89] M. Yang, K. Kpalma, and J. Ronsin. A survey of shape feature extraction techniques. *Pattern recognition*, pages 43–90, 2008.

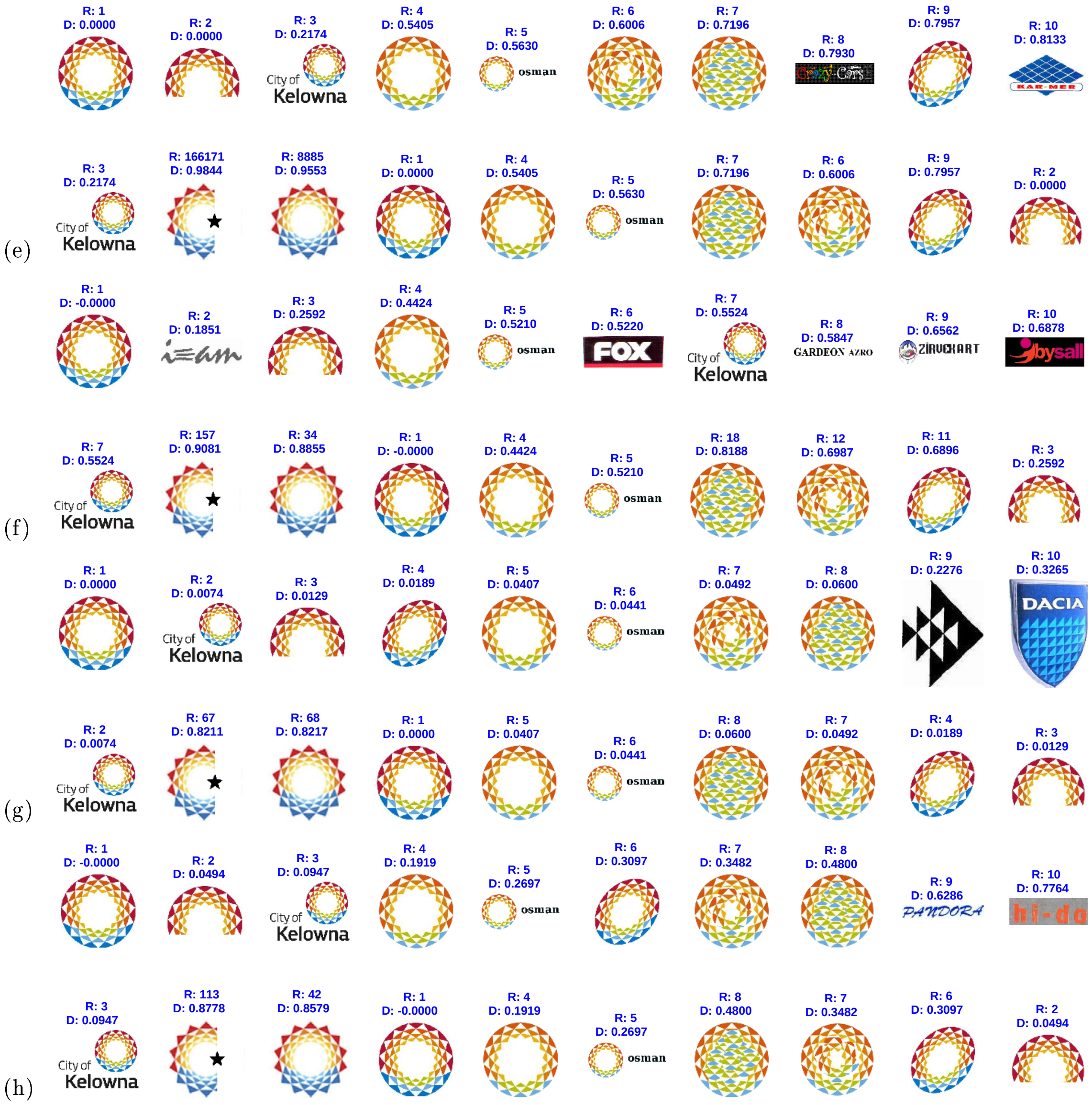
[90] C. Zhang and F. C. You. The technique of shape-based multi-feature combination of trademark image retrieval. *Advanced Materials Research*, 429:287–291, 2012.

[91] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.

APPENDIX A

In this appendix, we present visual retrieval results of methods implemented in our work.




R: 1 D: 0.0000
R: 2 D: 0.0000
R: 3 D: 0.2174
City of Kelowna
R: 4 D: 0.5405
R: 5 D: 0.5630
osman
R: 6 D: 0.6006
R: 7 D: 0.7196
R: 8 D: 0.7930
Crazy Cars
R: 9 D: 0.7957
R: 10 D: 0.8133
KAR-MER
(e)
R: 3 D: 0.2174
City of Kelowna
R: 166171 D: 0.9844
R: 8885 D: 0.9553
R: 1 D: 0.0000
R: 4 D: 0.5405
R: 5 D: 0.5630
osman
R: 7 D: 0.7196
R: 6 D: 0.6006
R: 9 D: 0.7957
R: 2 D: 0.0000
R: 1 D: -0.0000
R: 2 D: 0.1851
iżam
R: 3 D: 0.2592
R: 4 D: 0.4424
R: 5 D: 0.5210
osman
R: 6 D: 0.5220
FOX
R: 7 D: 0.5524
City of Kelowna
R: 8 D: 0.5847
GARDEON AZRO
R: 9 D: 0.6562
ZIRVEKART
R: 10 D: 0.6878
bysall
(f)
R: 7 D: 0.5524
City of Kelowna
R: 157 D: 0.9081
R: 34 D: 0.8855
R: 1 D: -0.0000
R: 4 D: 0.4424
R: 5 D: 0.5210
osman
R: 18 D: 0.8188
R: 12 D: 0.6987
R: 11 D: 0.6896
R: 3 D: 0.2592
R: 1 D: 0.0000
R: 2 D: 0.0074
City of Kelowna
R: 3 D: 0.0129
R: 4 D: 0.0189
R: 5 D: 0.0407
R: 6 D: 0.0441
osman
R: 7 D: 0.0492
R: 8 D: 0.0600
R: 9 D: 0.2276
R: 10 D: 0.3265
DACIA
(g)
R: 2 D: 0.0074
City of Kelowna
R: 67 D: 0.8211
R: 68 D: 0.8217
R: 1 D: 0.0000
R: 5 D: 0.0407
R: 6 D: 0.0441
osman
R: 8 D: 0.0600
R: 7 D: 0.0492
R: 4 D: 0.0189
R: 3 D: 0.0129
R: 1 D: -0.0000
R: 2 D: 0.0494
R: 3 D: 0.0947
City of Kelowna
R: 4 D: 0.1919
R: 5 D: 0.2697
osman
R: 6 D: 0.3097
R: 7 D: 0.3482
R: 8 D: 0.4800
R: 9 D: 0.6286
PANDORA
R: 10 D: 0.7764
hi-do
(h)
R: 3 D: 0.0947
City of Kelowna
R: 113 D: 0.8778
R: 42 D: 0.8579
R: 1 D: -0.0000
R: 4 D: 0.1919
R: 5 D: 0.2697
osman
R: 8 D: 0.4800
R: 7 D: 0.3482
R: 6 D: 0.3097
R: 2 D: 0.0494

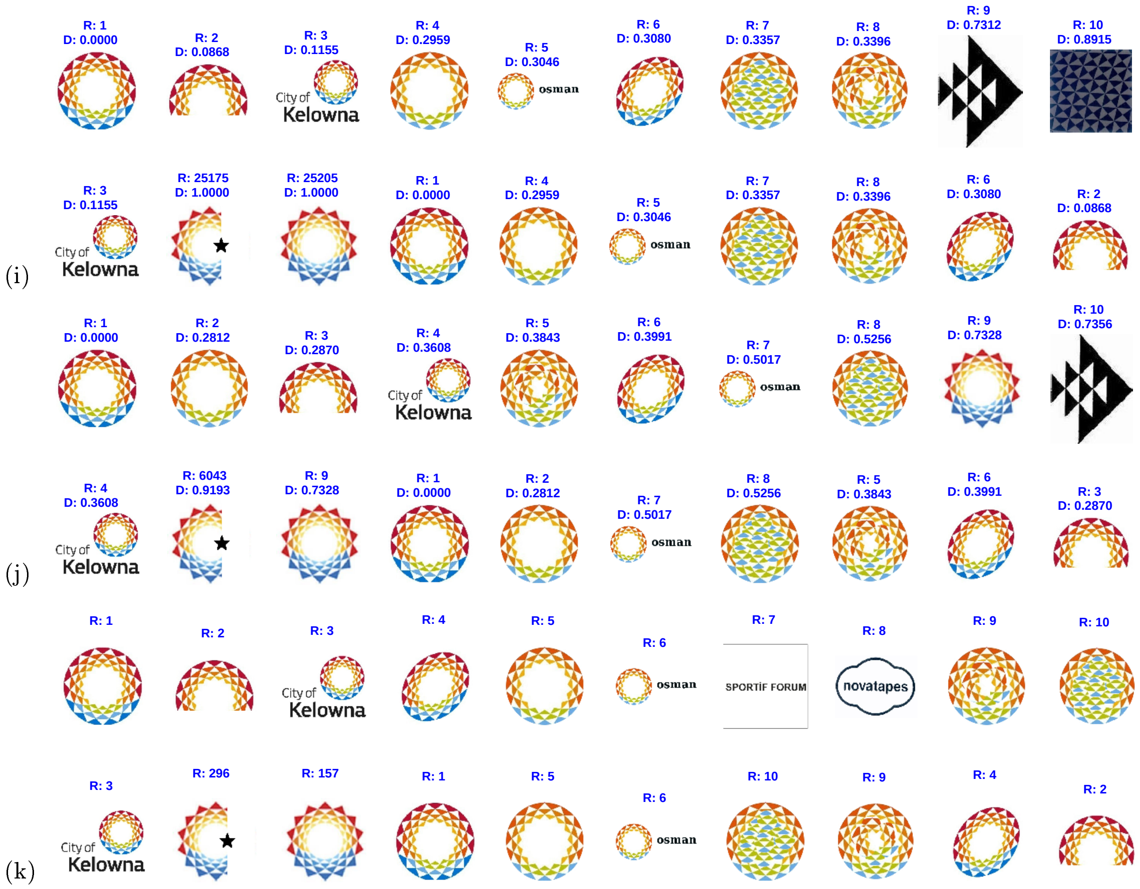
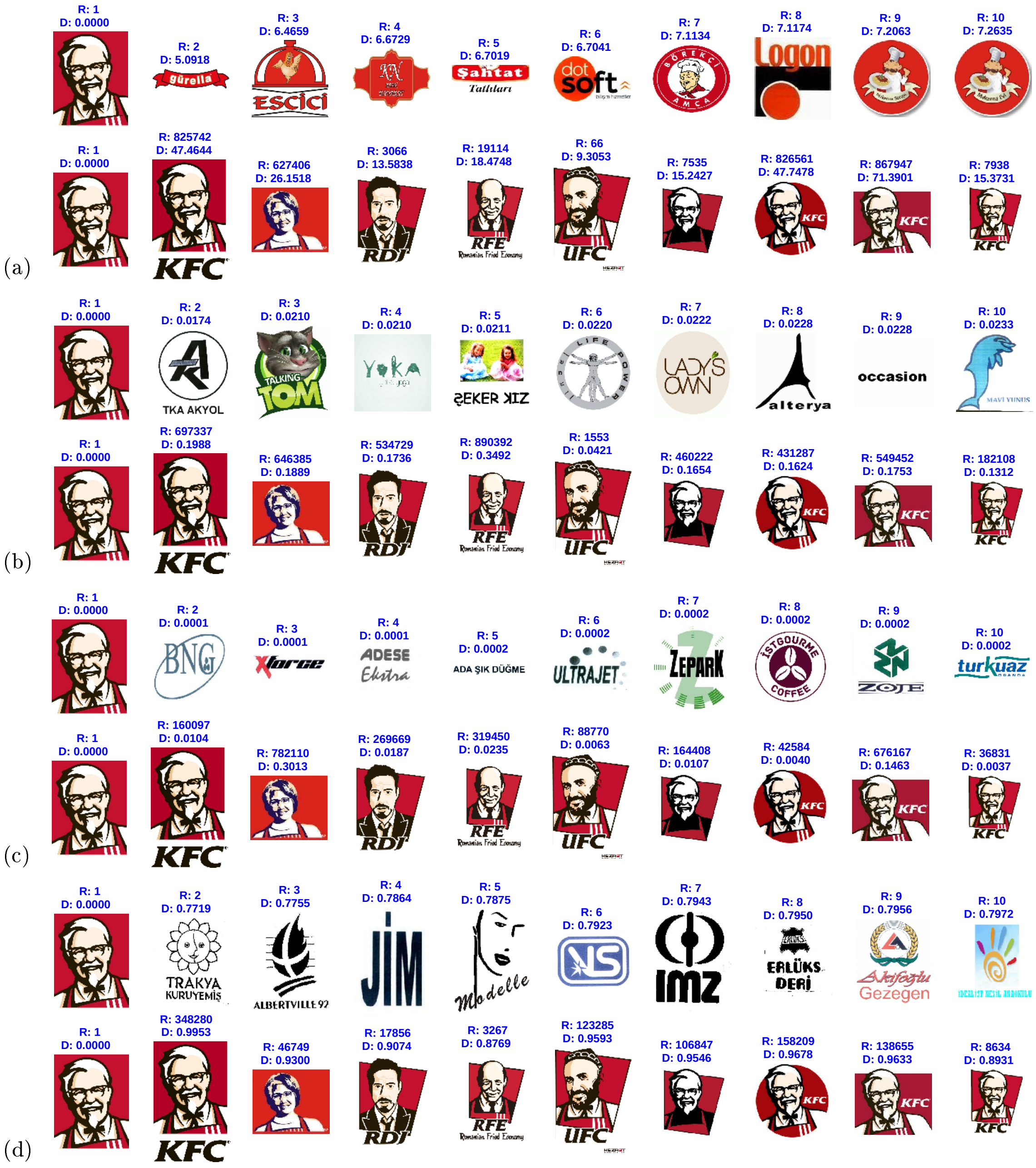
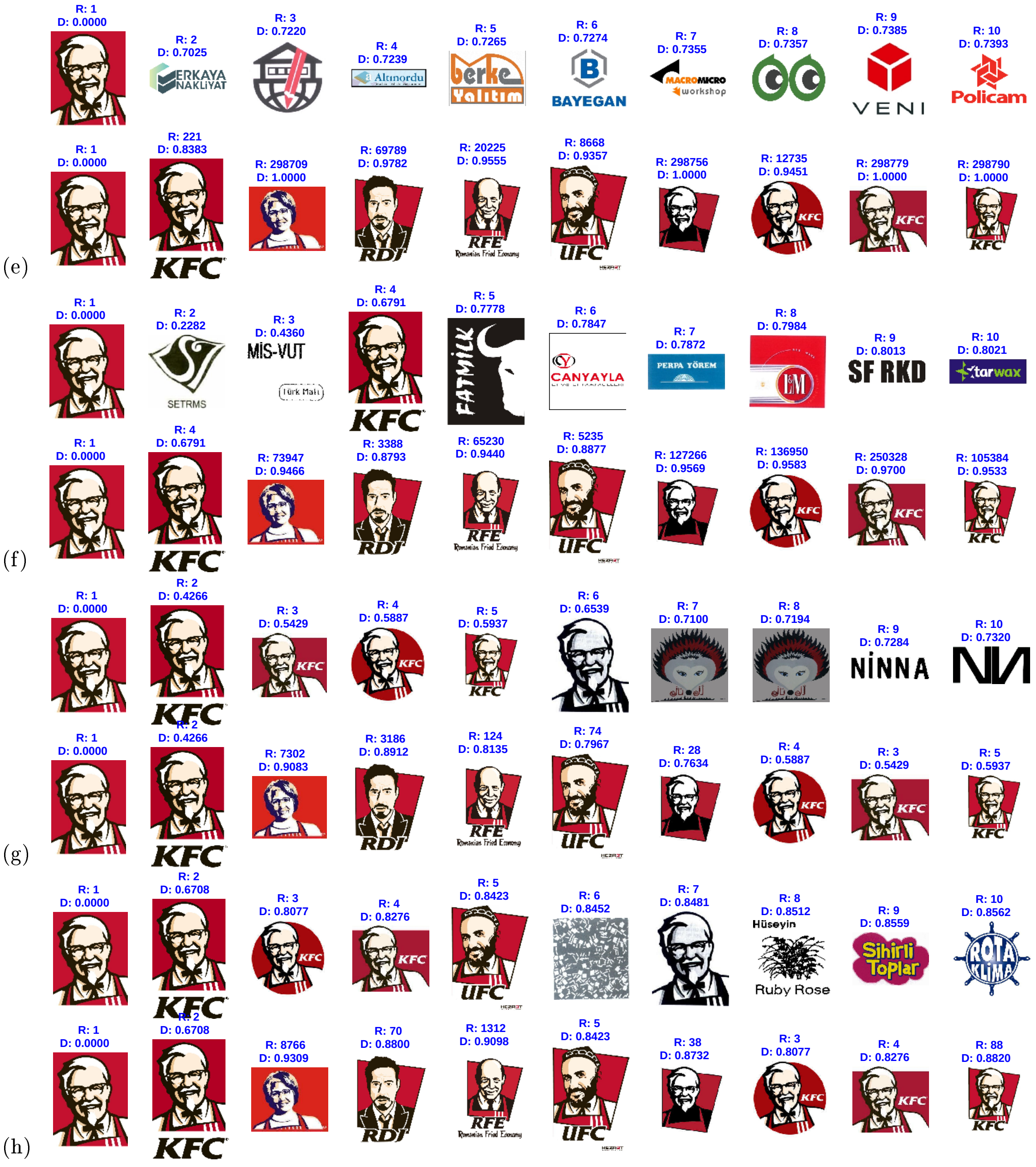


Figure A.-2: Sample retrieving results of various methods. These methods are: (a) Color, (b) Edge orientation, (c) LBP, d. Shapemes, (e) HOG. (f) ORB, (g) SIFT, (h) SURF, (i) Triangular SIFT, (j) Gradient descent. (k) IRP. In each result, the first row is the top 10 retrieved results of the first trademark in the first row of all results and the second row is the expected results. The D and R in titles of images represent similarity distance between query image and that image and ranking position.


(a)
R: 1 D: 0.0000
R: 2 D: 5.0918
R: 3 D: 6.4659
R: 4 D: 6.6729
R: 5 D: 6.7019
R: 6 D: 6.7041
R: 7 D: 7.1134
R: 8 D: 7.1174
R: 9 D: 7.2063
R: 10 D: 7.2635
R: 1 D: 0.0000
R: 825742 D: 47.4644
R: 627406 D: 26.1518
R: 3066 D: 13.5838
R: 19114 D: 18.4748
R: 66 D: 9.3053
R: 7535 D: 15.2427
R: 826561 D: 47.7478
R: 867947 D: 71.3901
R: 7938 D: 15.3731
(b)
R: 1 D: 0.0000
R: 2 D: 0.0174
R: 3 D: 0.0210
R: 4 D: 0.0210
R: 5 D: 0.0211
R: 6 D: 0.0220
R: 7 D: 0.0222
R: 8 D: 0.0228
R: 9 D: 0.0228
R: 10 D: 0.0233
R: 1 D: 0.0000
R: 697337 D: 0.1988
R: 646385 D: 0.1889
R: 534729 D: 0.1736
R: 890392 D: 0.3492
R: 1553 D: 0.0421
R: 460222 D: 0.1654
R: 431287 D: 0.1624
R: 549452 D: 0.1753
R: 182108 D: 0.1312
(c)
R: 1 D: 0.0000
R: 2 D: 0.0001
R: 3 D: 0.0001
R: 4 D: 0.0001
R: 5 D: 0.0002
R: 6 D: 0.0002
R: 7 D: 0.0002
R: 8 D: 0.0002
R: 9 D: 0.0002
R: 10 D: 0.0002
R: 1 D: 0.0000
R: 160097 D: 0.0104
R: 782110 D: 0.3013
R: 269669 D: 0.0187
R: 319450 D: 0.0235
R: 88770 D: 0.0063
R: 164408 D: 0.0107
R: 42584 D: 0.0040
R: 676167 D: 0.1463
R: 36831 D: 0.0037
(d)
R: 1 D: 0.0000
R: 2 D: 0.7719
R: 3 D: 0.7755
R: 4 D: 0.7864
R: 5 D: 0.7875
R: 6 D: 0.7923
R: 7 D: 0.7943
R: 8 D: 0.7950
R: 9 D: 0.7956
R: 10 D: 0.7972
R: 1 D: 0.0000
R: 348280 D: 0.9953
R: 46749 D: 0.9300
R: 17856 D: 0.9074
R: 3267 D: 0.8769
R: 123285 D: 0.9593
R: 106847 D: 0.9546
R: 158209 D: 0.9678
R: 138655 D: 0.9633
R: 8634 D: 0.8931



R: 1 D: 0.0000
R: 2 D: 0.7025
R: 3 D: 0.7220
R: 4 D: 0.7239
R: 5 D: 0.7265
R: 6 D: 0.7274
R: 7 D: 0.7355
R: 8 D: 0.7357
R: 9 D: 0.7385
R: 10 D: 0.7393
R: 1 D: 0.0000
R: 221 D: 0.8383
R: 298709 D: 1.0000
R: 69789 D: 0.9782
R: 20225 D: 0.9555
R: 8668 D: 0.9357
R: 298756 D: 1.0000
R: 12735 D: 0.9451
R: 298779 D: 1.0000
R: 298790 D: 1.0000
(e)
R: 1 D: 0.0000
R: 2 D: 0.2282
R: 3 D: 0.4360
R: 4 D: 0.6791
R: 5 D: 0.7778
R: 6 D: 0.7847
R: 7 D: 0.7872
R: 8 D: 0.7984
R: 9 D: 0.8013
R: 10 D: 0.8021
R: 1 D: 0.0000
R: 4 D: 0.6791
R: 73947 D: 0.9466
R: 3388 D: 0.8793
R: 65230 D: 0.9440
R: 5235 D: 0.8877
R: 127266 D: 0.9569
R: 136950 D: 0.9583
R: 250328 D: 0.9700
R: 105384 D: 0.9533
(f)
R: 1 D: 0.0000
R: 2 D: 0.4266
R: 3 D: 0.5429
R: 4 D: 0.5887
R: 5 D: 0.5937
R: 6 D: 0.6539
R: 7 D: 0.7100
R: 8 D: 0.7194
R: 9 D: 0.7284
R: 10 D: 0.7320
R: 1 D: 0.0000
R: 2 D: 0.4266
R: 7302 D: 0.9083
R: 3186 D: 0.8912
R: 124 D: 0.8135
R: 74 D: 0.7967
R: 28 D: 0.7634
R: 4 D: 0.5887
R: 3 D: 0.5429
R: 5 D: 0.5937
(g)
R: 1 D: 0.0000
R: 2 D: 0.6708
R: 3 D: 0.8077
R: 4 D: 0.8276
R: 5 D: 0.8423
R: 6 D: 0.8452
R: 7 D: 0.8481
R: 8 D: 0.8512
R: 9 D: 0.8559
R: 10 D: 0.8562
R: 1 D: 0.0000
R: 2 D: 0.6708
R: 8766 D: 0.9309
R: 70 D: 0.8800
R: 1312 D: 0.9098
R: 5 D: 0.8423
R: 38 D: 0.8732
R: 3 D: 0.8077
R: 4 D: 0.8276
R: 88 D: 0.8820
(h)

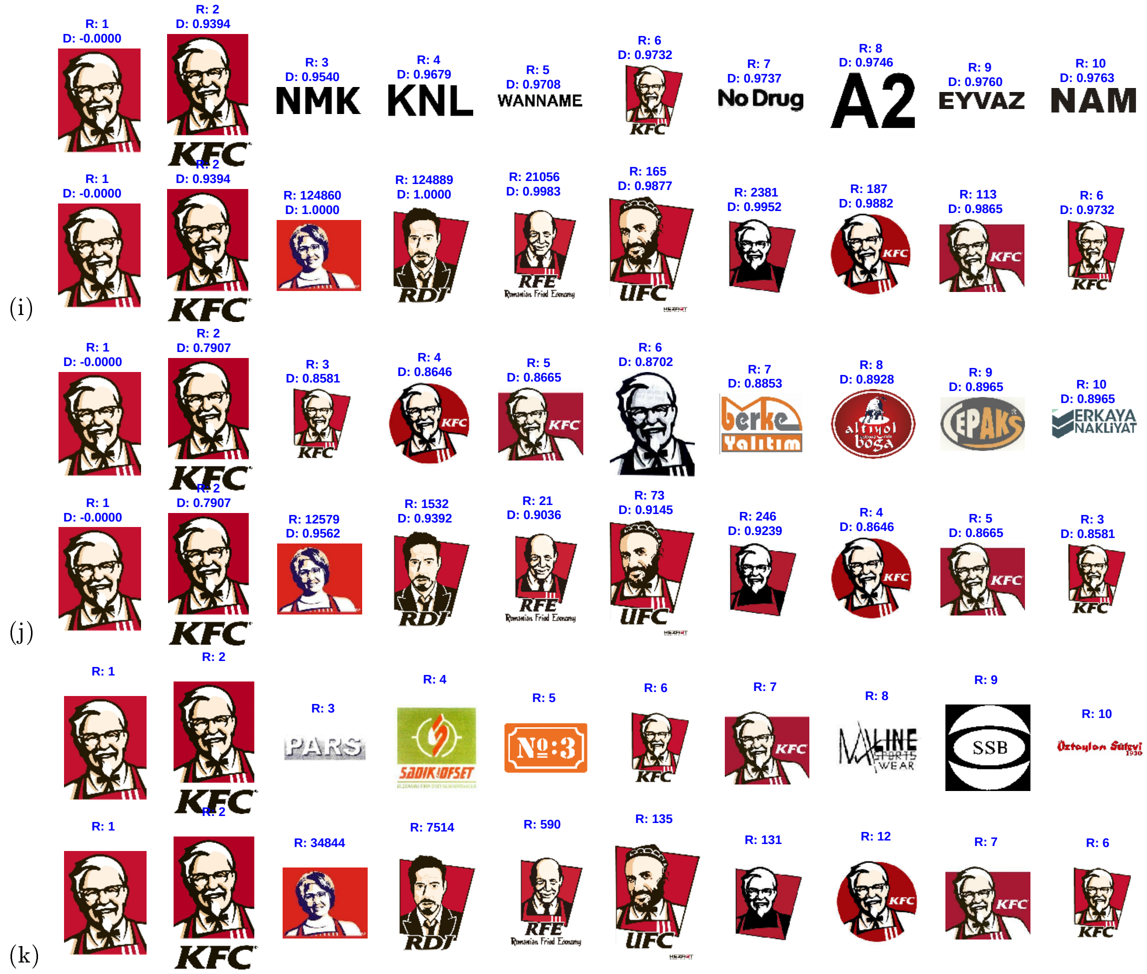
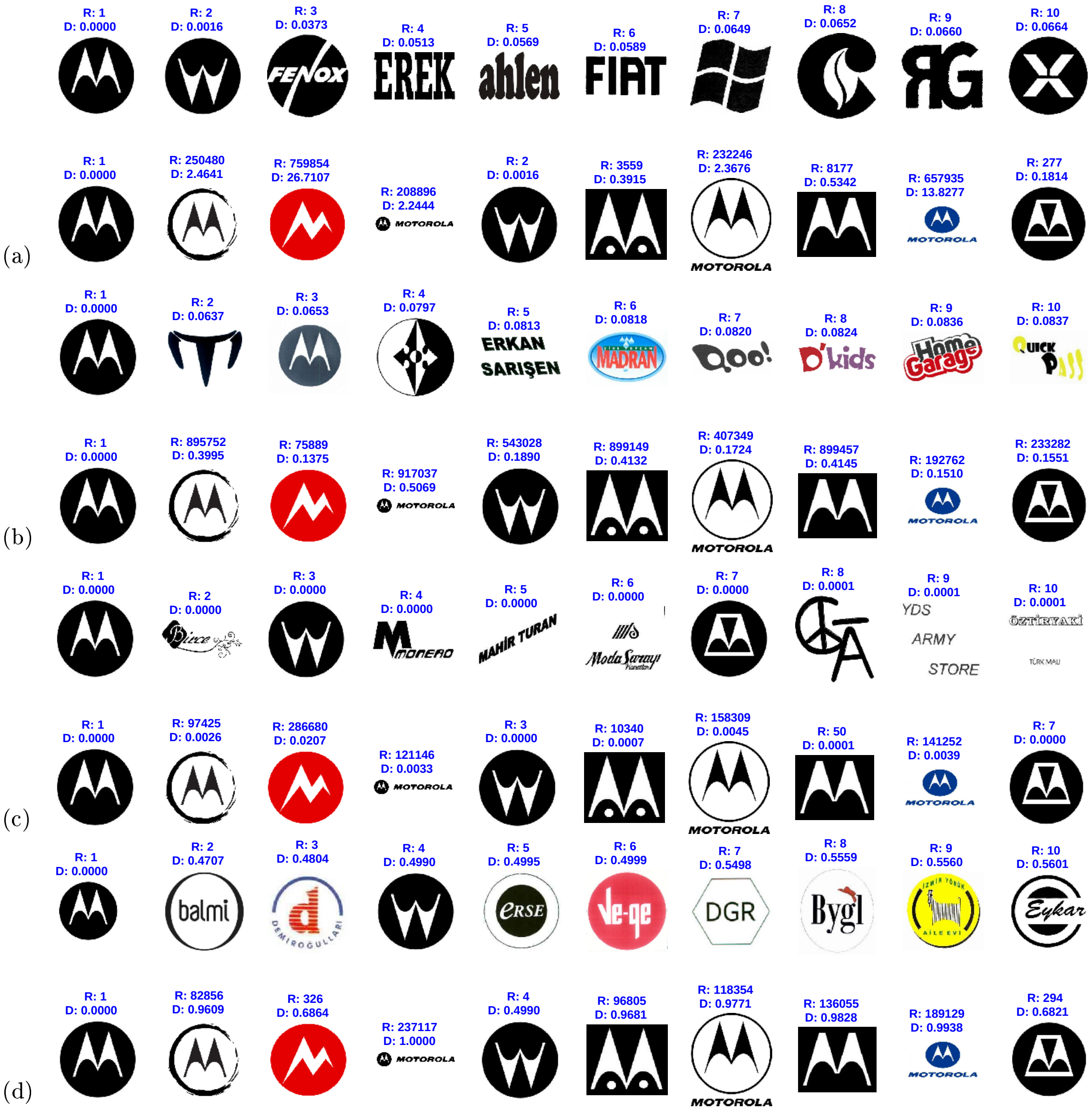
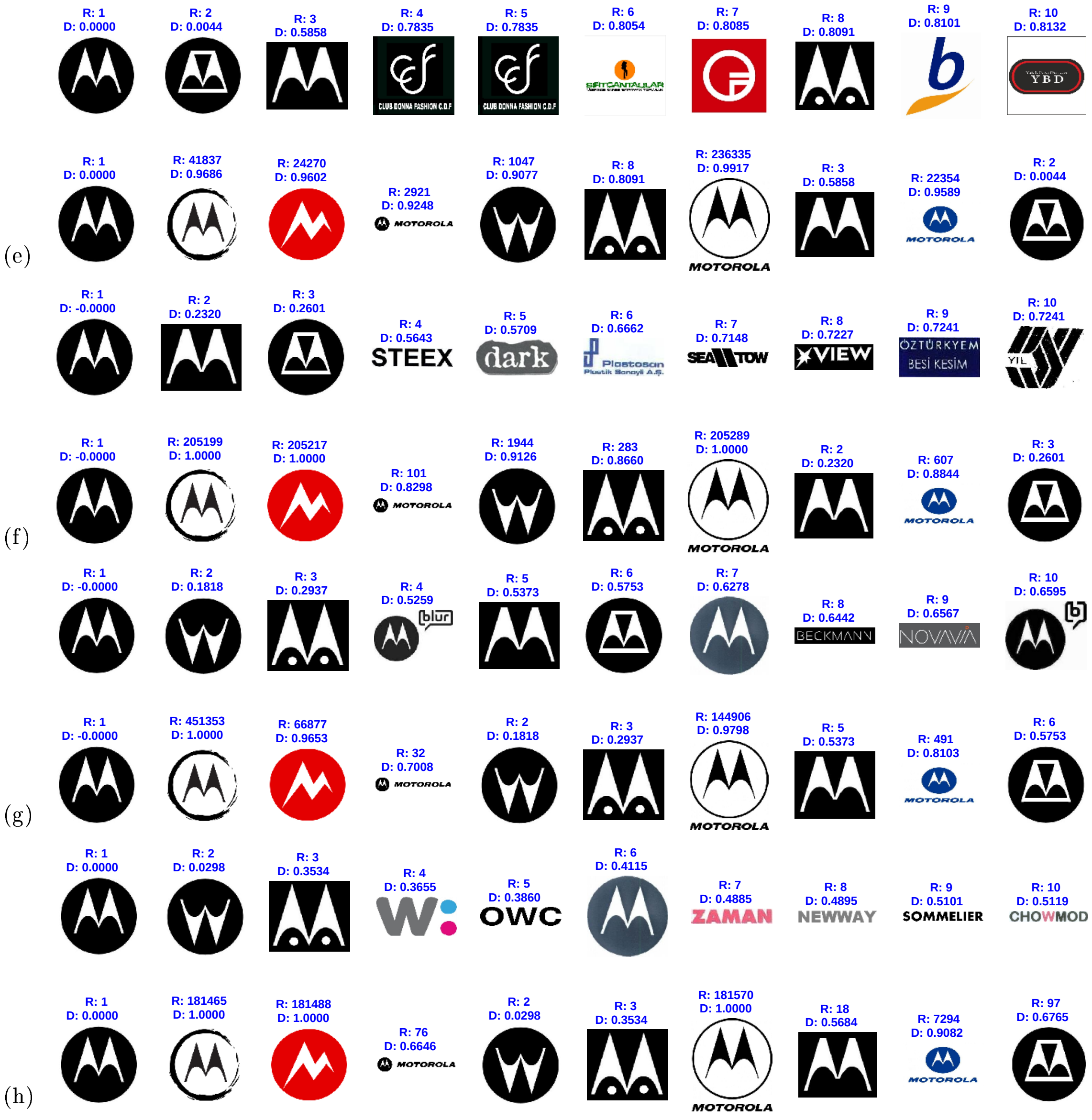


Figure A.-4: Sample retrieving results of various methods. These methods are: (a) Color, (b) Edge orientation, (c) LBP, d. Shapemes, (e) HOG. (f) ORB, (g) SIFT, (h) SURF, (i) Triangular SIFT, (j) Gradient descent. (k) IRP. In each result, the first row is the top 10 retrieved results of the first trademark in the first row of all results and the second row is the expected results. The D and R in titles of images represent similarity distance between query image and that image and ranking position.


(a)
R: 1 D: 0.0000
R: 2 D: 0.0016
R: 3 D: 0.0373
R: 4 D: 0.0513
R: 5 D: 0.0569
R: 6 D: 0.0589
R: 7 D: 0.0649
R: 8 D: 0.0652
R: 9 D: 0.0660
R: 10 D: 0.0664
R: 1 D: 0.0000
R: 250480 D: 2.4641
R: 759854 D: 26.7107
R: 208896 D: 2.2444
R: 2 D: 0.0016
R: 3559 D: 0.3915
R: 232246 D: 2.3676
R: 8177 D: 0.5342
R: 657935 D: 13.8277
R: 277 D: 0.1814
R: 1 D: 0.0000
R: 2 D: 0.0637
R: 3 D: 0.0653
R: 4 D: 0.0797
R: 5 D: 0.0813
R: 6 D: 0.0818
R: 7 D: 0.0820
R: 8 D: 0.0824
R: 9 D: 0.0836
R: 10 D: 0.0837
(b)
R: 1 D: 0.0000
R: 895752 D: 0.3995
R: 75889 D: 0.1375
R: 917037 D: 0.5069
R: 543028 D: 0.1890
R: 899149 D: 0.4132
R: 407349 D: 0.1724
R: 899457 D: 0.4145
R: 192762 D: 0.1510
R: 233282 D: 0.1551
R: 1 D: 0.0000
R: 2 D: 0.0000
R: 3 D: 0.0000
R: 4 D: 0.0000
R: 5 D: 0.0000
R: 6 D: 0.0000
R: 7 D: 0.0000
R: 8 D: 0.0001
R: 9 D: 0.0001
R: 10 D: 0.0001
(c)
R: 1 D: 0.0000
R: 97425 D: 0.0026
R: 286680 D: 0.0207
R: 121146 D: 0.0033
R: 3 D: 0.0000
R: 10340 D: 0.0007
R: 158309 D: 0.0045
R: 50 D: 0.0001
R: 141252 D: 0.0039
R: 7 D: 0.0000
R: 1 D: 0.0000
R: 2 D: 0.4707
R: 3 D: 0.4804
R: 4 D: 0.4990
R: 5 D: 0.4995
R: 6 D: 0.4999
R: 7 D: 0.5498
R: 8 D: 0.5559
R: 9 D: 0.5560
R: 10 D: 0.5601
(d)
R: 1 D: 0.0000
R: 82856 D: 0.9609
R: 326 D: 0.6864
R: 237117 D: 1.0000
R: 4 D: 0.4990
R: 96805 D: 0.9681
R: 118354 D: 0.9771
R: 136055 D: 0.9828
R: 189129 D: 0.9938
R: 294 D: 0.6821


R: 1 D: 0.0000
R: 2 D: 0.0044
R: 3 D: 0.5858
R: 4 D: 0.7835
CLUB DONNA FASHION C.D.F
R: 5 D: 0.7835
CLUB DONNA FASHION C.D.F
R: 6 D: 0.8054
R: 7 D: 0.8085
R: 8 D: 0.8091
R: 9 D: 0.8101
R: 10 D: 0.8132
YBD
(e)
R: 1 D: 0.0000
R: 41837 D: 0.9686
R: 24270 D: 0.9602
R: 2921 D: 0.9248
MOTOROLA
R: 1047 D: 0.9077
R: 8 D: 0.8091
R: 236335 D: 0.9917
MOTOROLA
R: 3 D: 0.5858
R: 22354 D: 0.9589
MOTOROLA
R: 2 D: 0.0044
R: 1 D: -0.0000
R: 2 D: 0.2320
R: 3 D: 0.2601
R: 4 D: 0.5643
STEEX
R: 5 D: 0.5709
dark
R: 6 D: 0.6662
Plastosan
R: 7 D: 0.7148
SEATOW
R: 8 D: 0.7227
VIEW
R: 9 D: 0.7241
ÖZTÜRKYEM
BESİ KESİM
R: 10 D: 0.7241
YIL
(f)
R: 1 D: -0.0000
R: 205199 D: 1.0000
R: 205217 D: 1.0000
R: 101 D: 0.8298
MOTOROLA
R: 1944 D: 0.9126
R: 283 D: 0.8660
R: 205289 D: 1.0000
MOTOROLA
R: 2 D: 0.2320
R: 607 D: 0.8844
MOTOROLA
R: 3 D: 0.2601
R: 1 D: -0.0000
R: 2 D: 0.1818
R: 3 D: 0.2937
R: 4 D: 0.5259
blur
R: 5 D: 0.5373
R: 6 D: 0.5753
R: 7 D: 0.6278
R: 8 D: 0.6442
BECKMANN
R: 9 D: 0.6567
NOVAVIA
R: 10 D: 0.6595
(g)
R: 1 D: -0.0000
R: 451353 D: 1.0000
R: 66877 D: 0.9653
R: 32 D: 0.7008
MOTOROLA
R: 2 D: 0.1818
R: 3 D: 0.2937
R: 144906 D: 0.9798
MOTOROLA
R: 5 D: 0.5373
R: 491 D: 0.8103
MOTOROLA
R: 6 D: 0.5753
R: 1 D: 0.0000
R: 2 D: 0.0298
R: 3 D: 0.3534
R: 4 D: 0.3655
R: 5 D: 0.3860
OWC
R: 6 D: 0.4115
R: 7 D: 0.4885
ZAMAN
R: 8 D: 0.4895
NEWWAY
R: 9 D: 0.5101
SOMMELIER
R: 10 D: 0.5119
CHOWMOD
(h)
R: 1 D: 0.0000
R: 181465 D: 1.0000
R: 181488 D: 1.0000
R: 76 D: 0.6646
MOTOROLA
R: 2 D: 0.0298
R: 3 D: 0.3534
R: 181570 D: 1.0000
MOTOROLA
R: 18 D: 0.5684
R: 7294 D: 0.9082
MOTOROLA
R: 97 D: 0.6765

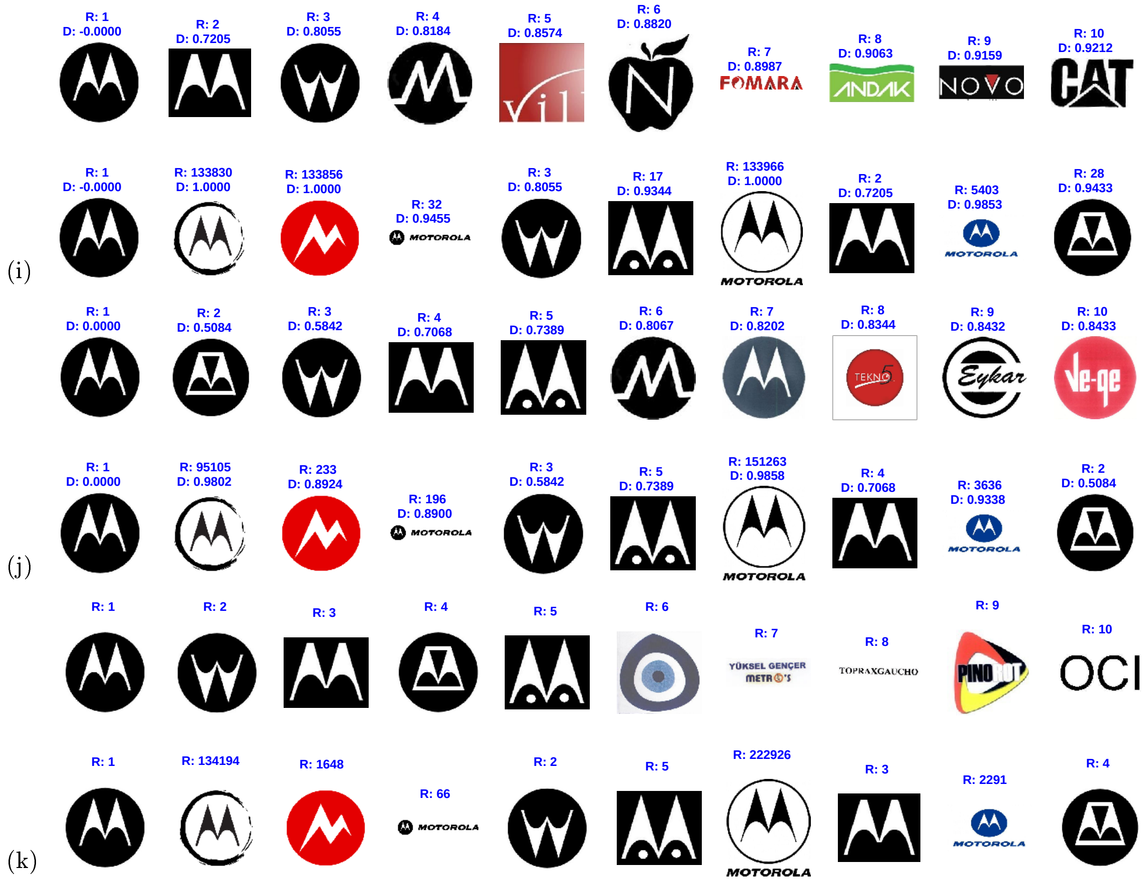


Figure A.-6: Sample retrieving results of various methods. These methods are: (a) Color, (b) Edge orientation, (c) LBP, d. Shapemes, (e) HOG. (f) ORB, (g) SIFT, (h) SURF, (i) Triangular SIFT, (j) Gradient descent. (k) IRP. In each result, the first row is the top 10 retrieved results of the first trademark in the first row of all results and the second row is the expected results. The D and R in titles of images represent similarity distance between query image and that image and ranking position.

APPENDIX B

In this appendix, we present the formulas of distance metrics and normalization methods related to our work.

B.1 Distance metrics

The method used for calculating the distance between vectors are called distance metric. The distance of features explained in Chapter 3 are calculated with these distance metrics.

Euclidean

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i^2 - q_i^2)}. \tag{B.1}$$

Cosine

$$d(\mathbf{p},\mathbf{q}) = \frac{\sum_{i=1}^{n}(p_i \cdot q_i)}{\|\mathbf{p}\| \cdot \|\mathbf{q}\|}. \tag{B.2}$$

Intersection (L1)

$$d(\mathbf{p},\mathbf{q}) = 1 - \frac{\sum_{i=1}^{n}\min(p_i, q_i)}{\min(\|\mathbf{p}\|, \|\mathbf{q}\|)}. \tag{B.3}$$

Intersection (L2)

$$d(\mathbf{p},\mathbf{q}) = 1 - \sqrt{\sum_{i=1}^{n}\min(p_i^2, q_i^2)}. \tag{B.4}$$

we modified the intersection distance with L2 norm for our request.

Quadratic

$$d(\mathbf{p},\mathbf{q}) = (\mathbf{p}-\mathbf{q})^t\mathbf{A}(\mathbf{p}-\mathbf{q}). \tag{B.5}$$

Manhattan

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)}. \tag{B.6}$$

B.2 Normalization

To calculate the distance of vectors at various scales, appropriate normalization methods are necessary. Here, we present the formulas of two normalization methods implemented in this work.

L1-norm

$$\mathbf{nh} = \frac{\mathbf{h}}{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{h(i)}}. \tag{B.7}$$

L2-norm

$$\mathbf{nh} = \frac{\mathbf{h}}{\sqrt{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{h(i)^2}}}. \tag{B.8}$$