



SEYREK TEMSİL İLE İÇERİK TABANLI GÖRÜNTÜ ERİŞİMİ

Ceyhun ÇELİK

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2017

Ceyhun ÇELİK tarafından hazırlanan “SEYREK TEMSİL İLE İÇERİK TABANLI GÖRÜNTÜ ERİŞİMİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Hasan Şakir BİLGE

Elektrik-Elektronik Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. Hasan OĞUL

Bilgisayar Mühendisliği Anabilim Dalı, Başkent Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Fırat HARDALAÇ

Elektrik-Elektronik Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Suat ÖZDEMİR

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Yrd. Doç. Dr. Mehmet Serdar GÜZEL

Bilgisayar Mühendisliği Anabilim Dalı, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 01/12/2017

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....
Prof. Dr. Hadi GÖKÇEN
Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
 - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
 - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
 - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
 - Bu tezde sunduğum çalışmanın özgün olduğunu,
- bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ceyhun ÇELİK

01 / 12 / 2017

SEYREK TEMSİL İLE İÇERİK TABANLI GÖRÜNTÜ ERİŞİMİ
(Doktora Tezi)

Ceyhun ÇELİK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Aralık 2017

ÖZET

İçerik Tabanlı Görüntü Erişimi (İTGE), görüntü ve çoklu ortam verilerinin artmasıyla birlikte, bu verilerden çıkarılan anlamlı bilgilerle arama işlemi yapmak ihtiyacı ile ortaya çıkmıştır. Günümüzde, Düşük Seviye Özelliklerden (DSÖ'lerden) Yüksek Seviye Anlamsal Bilgi'nin (YSAB) çıkarılmasında Yerel Özellik Tanımlayıcıları (YÖT'ları) kullanılmaktadır. Böylece, İTGE modelleri, Bölge Tabanlı Görüntü Erişimi (BTGE) yaklaşımları ile çözülmeye başlanmıştır. Seyrek Temsil (ST) uzun zamandır sinyal ve görüntü işleme yaklaşımlarında kullanılmakla birlikte son yıllarda İTGE problemlerine çözüm üretmektedir. Bu çalışmada ilk olarak literatürdeki YÖT ve ST algoritmalarının performansını değerlendirmek amacıyla bir İTGE çatısı önerilmiş ve bu çatı ile kapsamlı testler yapılmıştır. Daha sonra çok etiketli veri kümeleri için çevrimdışı ve çevrimiçi iki BTGE modeli önerilmiştir. Önerilen BTGE modellerin oluşturulma süresini hızlandırmak için de yeni bir YÖT kümeleme yaklaşımı önerilmiştir. Yapılan testlerde önerilen yaklaşımların literatürdeki yaklaşımlara göre daha başarılı sonuçlar verdiği gözlemlenmiştir.

Bilim Kodu : 92418
Bölge tabanlı görüntü erişimi, yerel özellik tanımlayıcıları,
Anahtar Kelimeler : bölümlenme, seyrek temsil
Sayfa Adedi : 95
Danışman : Doç. Dr. Hasan Şakir BİLGE

CONTENT BASED IMAGE RETRIEVAL WITH SPARSE REPRESENTATION

(Ph. D. Thesis)

Ceyhun ÇELİK

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

December 2017

ABSTRACT

Content Based Image Retrieval (CBIR) has emerged with the need to search with meaningful information extracted from these data, along with the increase in image and multimedia data. Today, Local Feature Descriptors (LFDs) are used to derive High Level Semantic Information (HLSI) from Low Level Features (LLFs). Thus, the CBIR models have begun to be solved through Region Based Image Retrieval (RBIR) approaches. Sparse Representation (SR) has been used in signal and image processing approaches for a long time and has been providing solutions to the CBIR problems in recent years. In this study, first, a CBIR framework is proposed in order to evaluate the performance of the LFDs and SR algorithms in the literature and comprehensive tests are made with this framework. Then, two offline and online RBIR models for multi-labeled data sets are proposed. A new LFD clustering approach is also adopted to speed up the building of recommended RBIR models. It has been observed that the proposed models are performed more successfully than the approaches in the literature.

Science Code : 92418
Key Words : Region based image retrieval, local feature descriptor, segmentation, sparse representation
Page Number : 95
Supervisor : Assoc. Prof. Dr. Hasan Şakir BİLGE

TEŐEKKÜR

Bu alıőmanın gerekleőtirilmesinde, 4 yıl boyunca deęerli bilgilerini benimle paylaőan, saygıdeęer danıőman hocam Do. Dr. Hasan Őakir BİLGE'ye, alıőma sūresince bana destek olan ve beni motive eden deęerli annem Cennet ELİK'e ve doktora eęitimim boyunca beni maddi olarak destekleyen TUBİTAK'a sonsuz teőekkūrlerimi sunarım. Ayrıca Photography veri tabanını baęıőlayan J. Li ve veri tabanını oluőturma konusundaki yardımlarından dolayı Dr. James Z. Wang'a teőekkūr ederim.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ	x
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	7
3. ÖZELLİK ÇIKARIMI VE SEÇİMİ	15
3.1. Yerel Özellik Tanımlayıcıları	15
3.1.1. SIFT	15
3.1.2. SURF	16
3.1.3. HoG	17
3.1.4. LBP ve LTP	17
3.2. Görüntü Bölümleme	18
4. SEYREK TEMSİL	23
4.1. Sözlük Öğrenme	25
4.2. Katsayı Öğrenme	26
4.2.1. OMP	27
4.2.2. Homotopy	28
4.2.3. LARS	28
4.2.4. Elastic net	29
4.2.5. Tekrarlayan daralma algoritmaları	29

	Sayfa
4.2.6. LLC	31
5. ÖNERİLEN YÖNTEMLER	33
5.1. İçerik Tabanlı Görüntü Erişim Çatısı	33
5.2. Yerel Özellik Tanımlayıcıları için Hızlandırılmış Görüntü Kümeleme	34
5.3. Çevrimdışı Bölge Tabanlı Görüntü Erişim Modeli	36
5.4. Çevrimiçi Bölge Tabanlı Görüntü Erişim Modeli	38
6. DENEYSEL SONUÇLAR	43
6.1. Veri Kümeleri ve Performans Değerlendirme Ölçütleri	43
6.1.1. Veri kümeleri	43
6.1.2. Performans değerlendirme ölçütleri	46
6.2. CBIR Çatısı Karşılaştırmalı Sonuçları	47
6.2.1. Parametreler	47
6.2.2. Coil 20 veri kümesi sonuçları	48
6.2.3. Corel 1000 veri kümesi sonuçları	53
6.3. Hızlandırılmış LFD Kümeleme Karşılaştırmalı Sonuçlar	59
6.3.1. Parametreler	59
6.3.2. Karşılaştırmalı sonuçlar	60
6.4. Çevrimdışı RBIR Modeli Sonuçları	63
6.4.1. Parametreler	63
6.4.2. Karşılaştırmalı sonuçlar	64
6.5. Çevrimiçi RBIR Modeli Sonuçları	69
6.5.1. Parametreler	70
6.5.2. Karşılaştırmalı sonuçlar	70
7. SONUÇ VE ÖNERİLER	79

	Sayfa
KAYNAKLAR	81
ÖZGEÇMİŞ	89
DİZİN	93



ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. CBIR yaklaşımlarında kullanılan yöntemler	12
Çizelge 2.2. SR tabanlı CBIR yaklaşımlarının performansları	13
Çizelge 4.1. OMP algoritması.....	27
Çizelge 4.2. Homotopy algoritması	28
Çizelge 4.3. SSF-SESOP	30
Çizelge 4.4. PCD-SESOP	31
Çizelge 6.1. Coil 20 üzerinde 512 boyutlu sözlükler için MAP değerleri.....	48
Çizelge 6.2. Coil 20 üzerinde K-SVD sözlük için MAP değerleri	49
Çizelge 6.3. Coil 20 üzerinde ODL algoritması için MAP değerleri.....	50
Çizelge 6.4. Coil 20 için tüm yaklaşımların en iyi sonuçları.....	50
Çizelge 6.5. Coil 20 için tüm sözlükler üzerinde katsayı öğrenme süreleri (saniye)....	52
Çizelge 6.6. Coil 20 için sözlük öğrenme süreleri (saniye)	53
Çizelge 6.7. Coil 20 için LFD algoritmaları süreleri (saniye)	53
Çizelge 6.8. Coil 20 için FBSR algoritması süreleri (saniye).....	53
Çizelge 6.9. Corel 1000 üzerinde 512 boyutlu sözlükler için MAP değerleri.....	54
Çizelge 6.10. Corel 1000 üzerinde K-SVD sözlük için MAP değerleri	55
Çizelge 6.11. Corel 1000 üzerinde ODL algoritması için MAP değerleri.....	55
Çizelge 6.12. Corel 1000 için tüm yaklaşımların en iyi sonuçları.....	56
Çizelge 6.13. Corel 1000 için katsayı öğrenme süreleri (saniye)	58
Çizelge 6.14. Corel 1000 için sözlük öğrenme süreleri (saniye)	58
Çizelge 6.15. Corel 1000 için LFD algoritmaları süreleri (saniye)	59
Çizelge 6.16. Corel 1000 için FBSR algoritması süreleri (saniye).....	59
Çizelge 6.17. Photography veri kümesi için kümeleme süreleri (dakika)	60

Çizelge	Sayfa
Çizelge 6.18. Photography veri kümesi için kümeleme benzerlikleri	61
Çizelge 6.19. ImageCLEF 2008 veri kümesi için bölümlenme süreleri (dakika)	62
Çizelge 6.20. ImageCLEF 2008 veri kümesi için kümeleme benzerlikleri	63
Çizelge 6.21. HoG için çevrimdışı RBIR sonuçları	64
Çizelge 6.22. HoG için çevrimdışı RBIR süreleri (dakika)	65
Çizelge 6.23. [8 8] LBP için çevrimdışı RBIR sonuçları	66
Çizelge 6.24. [8 8] LBP için çevrimdışı RBIR süreleri (dakika)	67
Çizelge 6.25. LTP için çevrimdışı RBIR sonuçları	68
Çizelge 6.26. LTP için çevrimdışı RBIR süreleri (dakika)	69
Çizelge 6.27. [8 8] LBP ve $t=0,5$ için çevrimiçi RBIR süreleri (dakika)	71
Çizelge 6.28. [8 8] LBP ve $t=0,5$ için çevrimiçi RBIR sonuçları	72
Çizelge 6.29. [4 4] LBP ve $t=0,5$ için çevrimiçi RBIR süreleri (dakika)	73
Çizelge 6.30. [4 4] LBP ve $t=0,5$ için çevrimiçi RBIR sonuçları	74
Çizelge 6.31. [4 4] LBP ve ortalama havuzlama için çevrimiçi RBIR sonuçları	75
Çizelge 6.32. Karşılaştırmalı sonuçlar	76

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 5.1. İçerik tabanlı görüntü erişimi çatısı.	34
Şekil 5.2. LFD kümeleme yaklaşımları	35
Şekil 5.3. Çevrimdışı RBIR modeli	38
Şekil 5.4. Çevrimiçi RBIR modeli.....	38
Şekil 6.1. Coil 20 veri tabanı objeleri	44
Şekil 6.2. Corel 1000 veri tabanı sınıfları	44
Şekil 6.3. Photography veri tabanı.....	45
Şekil 6.4. ImageCLEF 2008 veri tabanı.....	45
Şekil 6.5. Coil 20 için tüm yaklaşımların kesinlik-hatırlama eğrisi	51
Şekil 6.6. Coil 20 için tüm yaklaşımların CMC eğrisi.....	51
Şekil 6.7. Coil 20 veri kümesi üzerindeki rastgele 10 obje için en iyi ilk 20 sonuç.....	52
Şekil 6.8. Corel 1000 için tüm yaklaşımların kesinlik-hatırlama eğrisi	56
Şekil 6.9. Corel 1000 için tüm yaklaşımların CMC eğrisi.....	57
Şekil 6.10. Corel 1000 veri kümesi üzerindeki her bir sınıf için en iyi ilk 20 sonuç....	57

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

Kısaltmalar

Açıklamalar

AÇK	Alıcı Çalışma Karakteristiği
AGM	Aktif Görünüm Model
AIA	Automatic Image Annotation
AŞM	Aktif Şekil Model
ATF	Ayrılabilir Temsil Fonksiyonları
BDK	Bölge-kısıtlı Doğrusal Kodlama
BEK	Birikimli Eşleştirme Karakteristiği
BTGE	Bölge Tabanlı Görüntü Erişimi
CBIR	Content Based Image Retrieval
CBMIR	Content Based Medical Image Retrieval
CMC	Cumulative Match Characteristic
ÇB	Çizgi Bağlanımı
ÇÖİS	Çoklu Örnek İle Sorgulama
ÇSÖ	Çevrimiçi Sözlük Öğrenme
DoG	Difference of Gaussian
DOKO	Dereceli OKO
DSÖ	Düşük Seviye Özellikler
EKAB	En Küçük Açılı Bağlanım
EYY	En uygun Yönler Yöntemi
FBSR	Feature Based Sparse Representation
GF	Gauss Farkı
GİS	Görüntü İle Sorgulama
GMAP	Graded MAP
HGÖ	Hızlandırılmış Gürbüz Özellikler

Kısaltmalar**Açıklamalar**

HLSI	High Level Semantic Information
HoG	Histograms of Oriented Gradients
İTGE	İçerik Tabanlı Görüntü Erişimi
İTGME	İçerik Tabanlı Medikal Görüntü Erişimi
KPE	Mekânsal Piramit Eşleme
K-SVD	K- Singular Value Decomposition
K-TDA	K-Tekil Değer Ayırışma
LARS	Least-Angle Regression
LBP	Local Binary Pattern
LFD	Local Feature Descriptors
LLC	Locality-constrained Linear Coding
LLF	Low Level Features
LRGA	Local Regression and Global Alignment
LTP	Local Ternary Pattern
MAP	Mean Average Precision
MİS	Metin İle Sorgulama
MOD	Methods of Optimal Directions
NABK	Normalleştirilmiş Azaltılmış Birikimli Kazanç
NCDG	Normalized Discounted Cumulative Gain
ODL	Online Dictionary Learning
OET	Ortogonal Eşleşme Takibi
OGEA	Otomatik Görüntü Ek Açıklaması
OKO	Ortalama Kesinlik Ortalaması
OMP	Orthogonal Matching Pursuit
ÖBÖD	Ölçekten Bağımsız Özellik Dönüşümü
ÖTESK	ÖBÖD Tabanlı Elastic Seyrek Kodlama
ÖTST	Özellik Tabanlı Seyrek Temsil
PCD	Parallel Coordinate Descent
PKD	Paralel Koordinat Düşümü
QBI	Query By Image
QBME	Query By Multiple Example
QBT	Query By Text

Kısaltmalar**Açıklamalar**

RBIR	Region Based Image Retrieval
RR	Ridge Regression
SAO	Sıralı Altuzaylık Optimizasyonu
SBESC	SIFT Based Elastic Sparse Coding
SCBA	Sparse-Coding Based Approach
SESOP	Sequential Subspace Optimization
SIFT	Scale Invariant Feature Transform
SKTY	Seyrek Kodlama Tabanlı Yaklaşım
SPM	Spatial Pyramid Matching
SR	Sparse Representaiton
SSF	Seperable Surrogate Functionals
ST	Seyrek Temsil
SURF	Speeded-Up Robust Features
VN	Vektör Nicemleme
VQ	Vektor Quantization
YBKH	Yerel Bağlanım ve Küresel Hizalama
YGH	Yönlü Gradyan Histogramı
YİÖ	Yerel İkili Örüntü
YÖT	Yerel Özellik Tanımlayıcıları
YSAB	Yüksek Seviye Anlamsal Bilginin
YÜÖ	Yerel Üçlü Örüntü

1. GİRİŞ

Gelişen teknoloji ile birlikte özellikle internetin yaygınlaşması sonucunda insanların dijital ortamda sakladıkları ve paylaştıkları veriler büyük ölçüde artmıştır. Saklanan ve paylaşılan bilginin büyük bir çoğunluğunu görüntü ve videolar oluşturmaktadır. Artan bu çoğul ortam ve görüntü verilerine erişim ciddi bir problem haline gelmiştir. Bu verilerin erişimi için ilk yıllarda Query By Text (QBT) (Metin İle Sorgulama - MİS) yöntemleri kullanılmıştır [1-3]. Burada kullanıcı metin sorguları oluşturarak görüntüleri arar. Bu yöntemlerin uygulanabilmesi için tüm resimler ile ilgili açıklamaların olması gerekir. QBT yaklaşımlarının iki büyük problemi vardır. İlk problem, tüm görüntülere açıklama ekleme işlemi yapılmasıdır ve bu oldukça maliyetlidir. İkinci problem ise kullanıcıların görüntülere farklı anlamlar yükleyebilmesidir [3]. Böylece tüm görüntülere açıklama eklense de, kullanıcılar görüntülere farklı anlamlar yüklediği için, eklenen açıklamalar yetersiz olacaktır. Bu nedenle yeni bir yaklaşım olan Query By Image (QBI) (Görüntü İle Sorgulama - GİS) yöntemleri bu problemin çözümünde kullanılmıştır [1,3]. QBI yöntemleri diğer adıyla Content Based Image Retrieval (CBIR) (İçerik Tabanlı Görüntü Erişimi - İTGE) yöntemleri görüntü sorguları oluşturarak görüntüleri arar. Oluşturulan sorgularda görüntülerden çıkarılan Low Level Features (LLF) (Düşük Seviye Özellikler - DSÖ) kullanılarak High Level Semantic Information (HLSI) (Yüksek Seviye Anlamsal Bilginin - YSAB) elde edilmesi amaçlanmaktadır [4,5]. Burada kullanılan LLFs görüntüdeki renk, desen, şekil ve mekânsal yerleşim bilgileridir. CBIR yöntemleri bu özellikleri kullanarak resimler arasındaki anlamsal benzerliğin ortaya konulabileceğini, bir başka deyişle LLF ile HLSI arasında bir köprü oluşturulabileceğini varsayar. Ancak burada LLF ile HLSI arasında nasıl bir köprü oluşturulacağı büyük bir problemdir. Bu problem anlamsal boşluk olarak adlandırılır ve CBIR yaklaşımlarının çözmesi gereken en önemli problem bu anlamsal boşluğun giderilmesidir [6]. Görüntü erişimi yaklaşımları için literatürde yapılan adlandırma ile ilgili küçük farklılıklar mevcuttur. Zhang ve diğerleri QBT ve CBIR yaklaşımlarının yanında Automatic Image Annotation (AIA) (Otomatik Görüntü Ek Açıklaması - OGEA) yaklaşımlarını da üçüncü bir kategori olarak değerlendirir [7]. Bu modellerde görüntülere açıklamalar eklenir ve böylelikle QBT yaklaşımlarındaki gibi görüntüler değerlendirilebilir. Kalaivani ve diğerleri ise AIA yaklaşımlarını anlamsal boşluk problemini gidermek için kullanılacak bir yöntem olarak değerlendirirken bu yaklaşımları CBIR içerisinde AIA yaklaşımı olarak ele alırlar [8]. Bu

çalışmada görüntü erişimi yaklaşımları yalnızca QBT ve QBI olarak iki kategoride değerlendirilecektir.

CBIR modelleri üç temel adımdan oluşur. İlk olarak, özellik çıkarımı ve seçimi yapılır. Sonra, benzerlik ölçümü hesaplanır. Son olarak, indeksleme ve erişim gerçekleştirilir. Literatürde, bu adımları ayrıntılı olarak inceleyen birçok tarama ve değerlendirme çalışmaları bulunmaktadır [1-6, 9-13].

Özellik çıkarımı ve seçimi CBIR yaklaşımlarının temel adımlarından biridir. Bu işlem bölge tabanlı ya da tüm görüntü üzerinde gerçekleştirilebilir [13]. Görüntü tanımlayıcıları renk, desen, şekil ve mekânsal yerleşim bilgileri kullanılarak elde edilir ve görüntü erişimi işlemlerinde kullanılır. Literatürdeki çalışmalar incelendiğinde, CBIR yaklaşımları için son yıllarda yerel tanımlayıcıların kullanımının yoğun olduğu görülmüştür. Bir başka deyişle Region Based Image Retrieval (RBIR) (Bölge Tabanlı Görüntü Erişimi - BTGE) yaklaşımları kullanılmaktadır [3]. Buradaki yerel tanımlayıcılar, küresel tanımlayıcıların birebir aynısı olabilir, tek bir farkla ki bu tanımlayıcılar tüm görüntü yerine, görüntüdeki bölümlerden çıkarılır [3]. Bunun dışında tüm görüntü kullanılarak çıkarılan yerel tanımlayıcılar da vardır ve özellikle bu tanımlayıcılar CBIR yaklaşımlarında yoğun olarak kullanılmaktadır. Scale Invariant Feature Transform (SIFT) (Ölçekten Bağımsız Özellik Dönüşümü - ÖBÖD), Speeded-Up Robust Features (SURF) (Hızlandırılmış Gürbüz Özellikler - HGÖ), Histograms of Oriented Gradients (HoG) (Yönlü Gradyan Histogramı - YGH), Local Binary Pattern (LBP) (Yerel İkili Örüntü - YİÖ) CBIR yaklaşımlarında yoğun olarak kullanılan yerel tanımlayıcılardır [14 - 17]. Bunlara ek olarak LBP özelliklerinin geliştirilmiş sürümü Local Ternary Pattern (LTP) (Yerel Üçlü Örüntü - YÜÖ) tanımlayıcısı da yakın zamanda önerilen popüler bir Local Feature Descriptors (LFD) (Yerel Özellik Tanımlayıcıları - YÖT) yaklaşımıdır [18]. RBIR yaklaşımları ile CBIR yaklaşımları aynı adımları kullanır. Ancak RBIR yaklaşımları CBIR yaklaşımlarının aksine tüm görüntü yerine görüntüdeki bölgeleri kullanır. Bu nedenle RBIR yaklaşımları ile CBIR yaklaşımları arasındaki temel fark bölümlenme işleminin yapılmasıdır [3]. Görüntü bölümlenme yüksek seviye anlamsal bilgilerin elde edilmesinde etkili çözümler üretir. Özelliklerin bir arada sergilediği anlamlı bilginin kullanılmasını sağlamanın yanı sıra, görüntü üzerinde ilgisiz bölgelerin çıkarılmasını da sağlar. Bu nedenle birçok bilgisayarla görme probleminde önemli bir ön işleme adımı olarak kullanılır. Bölümlenme sayesinde görüntüdeki ilgisiz alanların çıkarılmasının yanı sıra temel eleman olarak

kullanılan bölgeler üzerinden resimler arasındaki benzerlik için daha hassas ölçümler gerçekleştirilir.

Benzerlik ölçümü ikinci adımdır. Bu adım CBIR modelleri için anahtar adımdır. Çünkü anlamsal boşluk bu adımda indirgenmeye çalışılır [3]. CBIR problemlerinin ilk zamanlarında, her bir özellik için farklı benzerlik ölçümüne ihtiyaç olduğu vurgulanmıştır [4]. Benzerlik ölçümü temel olarak görüntü tanımlayıcıları arasındaki uzaklığın ölçümüdür. Öklid ve Mahalanobis uzaklığı gibi temel ölçümler CBIR yaklaşımlarının ilk zamanlarındaki ölçümlerdir. Bununla birlikte, benzerlik ölçümü yerine benzerlik öğrenmesi yakın zamanda popüler olmuştur [3]. Dahası, makine öğrenmesi teknikleri de İTGE yaklaşımlarında kullanılmış, böylelikle karmaşık modeller önerilmiştir [7, 8, 19]. Sparse Representaiton (SR) (Seyrek Temsil – ST) bu amaçla son yıllarda görüntü erişimi üzerinde yoğun olarak çalışılmaktadır. SR yaklaşımının temel amacı giriş sinyallerinin daha basit bir birleşimle ifade edilmesidir. Bu nedenle giriş uzayı bir sözlüğe bağlı seyrek katsayılar ile ifade edilir. Bu özelliği ile çok boyutlu veri kümeleri üzerinde oldukça hızlı ve etkili çözümler üretir. Bu amaçla SR birçok bilgisayarla görme problemine uygulanmıştır [20, 21]. Sözlük öğrenme ve katsayı öğrenme olarak iki temel adımdan oluşur. Çevrimdışı yaklaşımlarda önce sözlük öğrenmesi yapılır, daha sonra öğrenilmiş sözlük üzerinde her bir giriş için seyrek katsayılar oluşturulur. Çevrimiçi yaklaşımlarda ise sözlük öğrenme ve katsayı öğrenme aynı anda gerçekleştirilir. Bu yaklaşımlarda ilk önce sözlük rastgele değerler ile belirlenir. Daha sonra seyrek katsayılar bu sözlüğe göre öğrenilir. Daha sonra elde edilen bu seyrek katsayılara göre sözlük öğrenimi yapılır ve bu işlem belirlenen sonlandırma ölçütüne kadar devam eder. SR kullanımı görüntü erişimi yaklaşımlarında oldukça yenidir [22].

Üçüncü ve son adımında ise indeksleme ve erişim işlemleri gerçekleştirilir. Yüksek boyutlu temsil edilen görüntüler üzerinde benzerlik araması yapmak, özellikle büyük ölçekli veri tabanları üzerinde oldukça maliyetli bir işlemdir. Literatürde indeksleme işlemi ağaç tabanlı, özet tabanlı ve görsel kelime tabanlı evrik indeks olarak üç kategoride incelenmektedir [23,24]. Ağaç ile oluşturulan veri yapıları birçok modelde etkili ve hızlı erişime olanak verir [25]. Ağaç tabanlı indeksleme vektör uzayında indeksleme ve ölçüm uzayında indeksleme olarak iki kategoride incelenir. Özellikler arasındaki benzerliği ölçmek için Öklid uzaklığını kullanan indeks yaklaşımları vektör uzayında indeksleme yaparken, Öklid uzaklığı kullanmayan indeks yaklaşımları ise ölçüm uzayında indeksleme

yapmaktadır [24]. Veri uzayını bölmesi açısından ise düzlemsel, küresel ve karma olarak üç kategoride incelenir [24]. Özet tabanlı indeksleme, ağaç tabanlı indeksleme ile kıyaslandığında etkili veri sıkıştırma ve indeksleme işlemi yapar [23]. Burada özellikler, özet fonksiyonları kullanılarak yüksek boyutlu uzaydan düşük boyutlu uzaya indirgenir. Böylece etkili veri sıkıştırma yapılarak erişim hızı artırılır. Özet tabanlı indeksleme Öklid uzaklığı kullanan tam Öklid bölge duyarlı özetleme ve Hamming uzaklığı kullanan spektral özet olarak iki kategoride incelenir [24]. Burada bölge duyarlı özet ile indeksleme oldukça popülerdir ve bu yöntem ile ilgili birçok iyileştirme mevcuttur [23]. Son olarak görsel kelime tabanlı evrik indeks yapıları vardır. Bu yapılar ilk olarak metin erişiminden üretilmiştir ve özellik kutusu ya da kelime kutusu olarak adlandırılır. Burada resim tanımlayıcıları indeksleme için en önemli adımdır. Tanımlayıcıların hızlı ve doğru bir şekilde ortaya konulması gerekir. Bununla beraber iki önemli adım daha vardır. Bunlar etkili bir evrik indeks yapısı ve tanımlayıcıların bilgisayar belleğinde saklanması için şifreleme ve sıkıştırma yöntemlerinin kullanılmasıdır [24].

Görüntülerin birden fazla etiketle etiketlenmesi işlemi ister QBT yönteminde ister CBIR yönteminde olsun, anlamsal boşluğun giderilmesi ve doğru sorgulama işlemi yapılması için oldukça önemlidir [8]. Çünkü bir görüntü her zaman birden fazla bilgiye sahip olabildiği gibi, farklı kullanıcılar için farklı anlamlar da içerebilir. Bu nedenle CBIR yaklaşımları için problemlerin çok etiketli olarak ele alınması ve modelin bu şekilde oluşturulması gerekir. CBIR ile ilgili yapılan incelemeler sonucunda tespit edilen problemler aşağıdaki şekildedir.

- CBIR yaklaşımlarındaki en temel problem ilk yıllarından bu yana anlamsal boşluk probleminin giderilmesidir.
- Anlamsal boşluğun giderilmesi konusunda önerilen öğrenme modelleri için yüksek boyutlu giriş uzayı ciddi bir problemdir ve bu nedenle etkili bir boyut indirgeme yaklaşımına ihtiyaç vardır.
- Eğitim işlemi için kullanılan doğru etiketli verinin az olması anlamsal boşluğun giderilmesi için önemli bir problemdir.
- Genel görüntü işleme problemlerinde kullanılan birçok görüntü tanımlayıcısından hangilerinin CBIR işlemleri için daha uygun olduğu tam olarak bilinmemektedir.
- Hızlı görüntü erişimi için etkili bir indeksleme işlemine ihtiyaç vardır.

Yakın zamanda yoğun olarak çalışılan CBIR yaklaşımları ise aşağıdaki gibidir.

- Anlamsal boşluk probleminin giderilmesi amacıyla benzerlik ölçümleri yerine benzerlik öğrenmesi algoritmalarının kullanılması.
- Anlamsal boşluk probleminin giderilmesi amacıyla görüntülerin tek etiket yerine çok etiketli ele alınması.
- Yeterli eğitim kümesi oluşturabilmek amacıyla etiketsiz görüntü kümesinin de öğrenme işleminde kullanılması.
- Küresel görüntü tanımlayıcılar yerine yerel tanımlayıcıların kullanılması.
- Görüntünün tamamının değerlendirilmesi yerine bölgesel olarak değerlendirilmesi.

Yukarıda tespit edilen problemlerin ve literatürde yapılan çalışmaların ışığında bu çalışmada öncelikle genel bir CBIR çatısı önerilmiş ve bu çatı üzerinden literatürde popüler olarak kullanılan algoritmaların başarımı karşılaştırılmıştır. Elde edilen sonuçlar literatürdeki SR tabanlı CBIR yaklaşımları ile karşılaştırılmıştır. Karşılaştırma işlemi Coil 20 [26] ve Corel 1000 [27] isimli iki tek etiketli veri kümesi üzerinde test edilmiştir. Daha sonra gözlemlenen sonuçlara göre başarılı bulunan algoritmalar kullanılarak iki RBIR yaklaşımı önerilmiştir. Önerilen yaklaşımlarda LFD tüm görüntü üzerinden çıkarıldıktan sonra görüntüdeki bölgelere göre kümelenmektedir. Bu nedenle kümeleme aşamasını hızlandırmak için bölümlenme aşaması için de yeni bir yaklaşım önerilmiştir. Bölümlenme için önerilen yaklaşım kümeleme benzerlikleri üzerinden standart bölümlenme algoritmaları ile kıyaslandıktan sonra, RBIR modellerinde de karşılaştırmalı olarak kullanılarak bu yaklaşımların görüntü erişimi başarımları da sınanmıştır. Önerilen yaklaşımlar bölge tabanlı oldukları için ve erişim için görüntülerin çok etiketli ele alınmasının daha hassas erişim ölçümü vereceğinden, çok etiketli Photography [28] ve ImageCLEF 2008 [29] veri kümeleri karşılaştırma için kullanılmıştır. Çalışmada LFD, RBIR yaklaşımları için oldukça başarılı sonuçlar verdiği için, SR ise yüksek boyutlu veriler üzerinde çalışırken verileri daha basit ifadelerle ifade ettiğinden ve RBIR için benzerlik öğrenmesi ihtiyacını giderebileceği için tercih edilmiştir. Yapılan çalışmanın literatüre özgün katkıları şu şekilde özetlenebilir:

- Bu çalışmada ilk olarak LFD ve SR kullanan CBIR yaklaşımları ayrıntılı olarak incelenmiştir. Yapılan inceleme sonucunda popüler olarak kullanılan algoritmalar

önerilen bir RBIR çatısı üzerinde test edilmiş ve algoritmaların karşılaştırmalı sonuçları incelenmiştir. Önerilen RBIR yaklaşımlarında bir önceki çatıda elde edilen sonuçlara göre başarılı olduğu gözlemlenen LFD, sözlük öğrenme ve katsayı öğrenme algoritmaları kullanılmıştır.

- Önerilen iki SR tabanlı RBIR yaklaşımında da bölümlene işlemi kullanılmıştır. Görüntüden çıkarılan LFD tüm görüntü üzerinden çıkarıldıktan sonra görüntü üzerindeki bölgelere göre kümeleneştir. Böylece SR ile erişim modeli oluşturulmadan önce kullanılan temel eleman tüm görüntü yerine görüntü bölgeleridir.
- LFD kümelemek için standart bölümlene algoritmalarının yanı sıra daha hızlı bir kümeleme algoritması önerilmiş böylece kümeleme işlemi hızlandırılmıştır.
- Kümelene LFD üzerinde veri kümesi boyutunu hızlandırmak için özellik seçilimi işlemine ek olarak özelliklerin ortalama ve varyans değerleri kullanılmıştır. Böylece modelin oluşturulma süresi büyük ölçüde hızlandırılmıştır.
- Erişim problemi çok etiketli olarak ele alınmış ve veri kümesindeki her bir etiket sözlük üzerinde bir atom ile temsil edilmiştir. Düşük sözlük boyutu erişim işlemini hızlandırmıştır.
- Önerilen yaklaşımlarda literatürde daha önce görüntü erişiminde kullanılmamış olan katsayı öğrenme algoritması kullanılmıştır. Buna ek olarak popüler olarak kullanılan çevrimiçi sözlük öğrenme algoritmasının etiketli sürümü önerilmiştir.

2. LİTERATÜR TARAMASI

Literatürde SR ile CBIR yaklaşımları son yıllarda yoğun olarak çalışılmıştır. Konu ile ilgili yayınların tarihlerine bakıldığında çalışmaların 2009-2017 yılları arasında yapıldığı görülmektedir. Bununla birlikte bu çalışmaların büyük çoğunluğu son birkaç yıl içerisinde yapılmıştır. SR algoritmalarında temel amaç giriş sinyallerinin daha basit bir birleşimle ifade edilmesidir ve giriş uzayının bir sözlüğe bağlı seyrek katsayılarla ifade edilmesidir. Daha önce bahsedildiği üzere birçok bilgisayarla görme problemine uygulanan SR yaklaşımının CBIR yaklaşımlarına nasıl uyarlanacağı tam olarak açık değildir [30]. Bu nedenle literatürdeki yaklaşımlarda SR farklı amaçlarla kullanılmıştır. Bazı çalışmalarda standart SR doğrudan görüntülerin temsili için kullanılmıştır [31-36].

Liu ve diğerleri yapmış oldukları çalışmada SR için transfer öğrenmesini anahtar kodlama öğrenme adı ile gerçekleştirmiştir [31]. Buradaki anahtar kelimesi, kodlamada resim parçalarının kullanımı yerine SURF tanımlayıcılarının kullanılmasından gelir. İlk olarak SR ile etiketli olmayan görüntü kümesinden sözlük ve katsayılar elde edilir. Sonrasında sözlük topografı, alt uzayda kümelenir. Daha sonra etiketli veriler katsayıların eniyilenmesinde kullanılır. Önerilen yaklaşım hem sınıflandırma hem de erişim problemleri için test edilmiş ve karşılaştırılmıştır. Erişim işlemi için, renk momenti ve kelime kutusu modelleri ile karşılaştırılmıştır. Önerilen yöntem daha az katman kullandığından daha iyi sonuçlar vermiştir. Kang ve diğerleri SIFT tanımlayıcılarını resimler üzerindeki döndürme, taşıma çevirme ya da bakış açısı değiştirme gibi işlemlere dayanıklı olması nedeni ile temel şema olarak belirlemişlerdir [32]. Feature Based Sparse Representation (FBSR) (Özellik Tabanlı Seyrek Temsil – ÖTST) isimli yaklaşımlarında hem test resmi hem de referans resimleri için SR ile sözlükler oluşturulmaktadır. Daha sonra bu sözlükler birleştirilerek test resminin bu sözlüğe göre katsayısı yeniden SR ile oluşturulur. Yeni oluşturulan katsayının atomları hangi sözlükten seçtiğine bakılarak iki resim arasındaki benzerlik değerlendirilir. Önerilen yaklaşım birçok görüntü işleme problemi için test edilmiştir ve CBIR problemi için görsel olarak belirli nokta özellikleri olarak açıklanmış bir yaklaşım ile karşılaştırılmış ve başarılı olduğu gözlemlenmiştir. Zhao ve diğerleri iki yönlü SR ile etiketli resimlerin sayısını arttırmayı amaçlayan işbirlikçi yarı gözetimli bir yaklaşım önermişlerdir [33]. Önerilen yaklaşımda SR, etiketli eğitim kümesindeki verilerle etiketsiz verileri etiketlemek için kullanan ileri yönlü SR olarak

adlandırılmıştır. Literatürde bu işlemin tersi olacak olan geri yönlü SR yaklaşımı olmadığından bu iki yöntemi birlikte kullanmışlardır. Yazarlara göre çekirdek tabanlı SR oldukça başarılı sonuçlar vermektedir. Bu nedenle çekirdek tabanlı SR, standart SR yaklaşımının yanı sıra bu iki yönlü algorithmada kullanılmıştır. Önerilen yaklaşımın sürümleri literatürde yakın zamanda önerilmiş olan üç yarı gözetimli yöntem ile karşılaştırılmıştır ve ilk veri kümesi için önerilen yaklaşım en iyi sonuçları verirken, ikinci veri kümesi için etiketli verinin %20 ve daha fazla olduğu durumlarda en iyi sonuçları verdiği gözlemlenmiştir. Thiagarajan ve diğerleri hem yerel hem de küresel resim özelliklerini kullanarak SR ile CBIR modeli önermişlerdir [34]. Önerilen yaklaşımda GIST küresel tanımlayıcı ile HoG ve LBP yerel tanımlayıcıları giriş vektörleri olarak kullanılmıştır. Önerilen yaklaşım SIFT + Spatial Pyramid Matching (SPM) (Mekânsal Piramit Eşleme - KPE) yöntemi ile karşılaştırılmış ve daha başarılı sonuçlar verdiği gözlemlenmiştir.

CBIR işlemlerindeki bazı uygulamalarda Query By Multiple Example (QBME) (Çoklu Örnek İle Sorgulama - ÇÖİS) işlemleri yapılmaktadır. Hong ve Zhu SR ile görüntü erişimi önermişlerdir [35]. Önerilen yöntemde ilk olarak resimler arasındaki anlamsal ilişki araştırılmıştır. Burada Anlamsal Korelasyon Çoklu Çizgesi resimleri modellemek için yapılandırılmıştır. İkinci yapılan iyileştirme ise çoklu örnekler ile yapılan sorgulamaların aynı anda yürütülmesi ile sorgulama süresinin azaltılmasıdır. Burada kullanılan çoklu çizge ile görüntülerin paylaştığı nesnelerin yanı sıra bu nesnelerin neler olduğu bilgisi de tutulmaktadır. Önerilen yöntem iki veri kümesi üzerinde test edilmiştir. İlk veri kümesi için Local Regression and Global Alignment (LRGA) (Yerel Bağlanım ve Küresel Hizalama - YBKH) yöntemiyle, ikinci veri kümesi için ise hem LRGA hem de L1 çizge temsili yöntemleriyle karşılaştırılmıştır. Önerilen yaklaşım hem daha yüksek kesinlik hem de daha az hesaplama zamanı ile en iyi sonuçları vermiştir. Ge ve diğerleri CBIR yaklaşımı olarak birden fazla özelliği sıkıştırmak için SR kullanmışlardır [36]. Çalışmada öncelikle her bir özellik tanımlayıcı seyrek kodlara dönüştürülür ve daha sonra bu kodlar birleştirilerek tek bir vektör olarak resmi temsil eder. Burada kodlar birleştirilirken her bir koddaki en büyük değer ya da kod vektörünün ortalaması alınarak skalar bir değer olarak görüntü vektörüne eklenir. Buradaki amaç ayrıştırıcı özelliklerin seçilmesi ve görüntü erişimi sırasında önemliliklerini vurgulamaktır.

Standart SR temsili ile yapılan CBIR yaklaşımlarına bakıldığında yerel tanımlayıcı kullanımı oldukça yaygındır [31-34, 36]. Standart SR ile yapılan CBIR yaklaşımlarının özgünlüğü benzer resimlerin sözlük üzerinde konumlandırılmaları, eğitim stratejisi, benzerlik ölçümünden kaynaklanır. Çalışma [31], [32] ve [33]'de her bir yerel tanımlayıcı için elde edilen katsayıların toplamı enküçüklemeye çalışılmaktadır. Çalışma [34] ve [36]'da enbüyük havuzlama ile her bir özellik tanımlayıcısı için elde edilen katsayılar birleştirilmekte ve tek bir vektör ile temsil edilmektedir. Çalışma [35]'de ise SR, çizgelerin oluşturulmasında kullanılmıştır. Bu çalışmalarda genel olarak görüntü etiketlerinden yararlanılmamıştır [31, 32, 34, 36].

SR ile CBIR yaklaşımlarında standart SR kullanımının yanı sıra, SR temsiline kullanılan enküçültme probleminin geliştirilmesinin yapıldığı çalışmalar da mevcuttur. [30, 37-48]. Huang ve diğerleri yapmış oldukları çalışmada kıyafet resimleri erişimi için bölge-duyarlı SR kullanmışlardır [30]. Önerdikleri yaklaşım Sparse-Coding Based Approach (SCBA) (Seyrek Kodlama Tabanlı Yaklaşım – SKTY) düşük seviye görüntü özelliklerinin yanı sıra yüksek seviye özellikleri de giriş uzayında kullanır. Yüksek seviye özellikler olarak kıyafetlerle ilgili yaka tipi, kol uzunluğu gibi özellikler kullanılırken, düşük seviye özellikler olarak CIE-Lab renk uzayı ile Gabor filtreleri, LBP ve SIFT kullanılmıştır. Yazarlar SR ve Locality-constrained Linear Coding (LLC) (Bölge-kısıtlı Doğrusal Kodlama – BDK) [49] kullanımının diğer görüntü işleme problemlerinde etkin kullanımlarından esinlenerek bu yöntemleri CBIR problemi üzerinde uygulamışlardır. Onlara göre burada eniyilemesi gereken problem Eş. 2.1'dedir.

$$\|x_i - D\alpha_i\|_2^2 + \lambda \|P_i \circ \alpha_i\|_2^2 + \eta(I^T \alpha_i - 1) \quad (2.1)$$

Burada standart SR enküçültme problemi yerel adaptör ile geliştirilerek bölge-duyarlı seyrek kodlama geliştirilmiştir. Önerilen yaklaşım standart benzerlik ölçümü yaklaşımları ile kıyaslanmış ve bu yöntemlerden daha başarılı olduğu gözlemlenmiştir. Sonraki çalışmalarında yazarlar önerdikleri yöntemi aktif öğrenme stratejisi ile geliştirmişlerdir [37]. Bir önceki çalışmada önerilen algoritma Destek Vektör Bağlanım ve kullanıcı etkileşimi ile eğitilmiştir. Önerilen yaklaşım literatürdeki çalışmalarla kıyaslandığında en iyi memnuniyet oranı elde edilmiştir.

Qiao ve Beling yapmış oldukları çalışmada çoklu örnek öğrenimi için geliştirilmiş SR kullanmışlardır [38]. Çoklu örneklerde sınıf etiketleri bir örnek yerine birden fazla örneğin organizasyonu ile oluşturulan kutulara atanır. Buradaki örnekler resimlerin bölümlenmesi sonucu oluşan yerel bölgelerdir. Önerilen yaklaşımda etiketsiz örneklerin bilgileri de eğitim işlemine aktarılarak transfer öğrenmesi işlemi gerçekleştirilmiştir. Çalışmada SR etiketler arasındaki veri dağılımını bölümlendirmek için kullanılmıştır. Bu nedenle hem ayrıştırıcı fonksiyon hem de sözlük ve örnek katsayılarının enküçültme problemi bir arada kullanılmıştır.

$$\min_{\{\alpha, \Theta\}} L(\alpha, f) + \lambda_0 \sum_{i,j} \|B_{ij}^l - D\alpha_{ij}\|_2^2 + \lambda_1 \sum_{i,j} |\alpha_{ij}| + \lambda_2 \sum_{i,j} \|\Theta\|_2^2 \quad (2.2)$$

Çalışmada ayrıştırıcı olarak doğrusal bağlanım, logaritmik bağlanım ve Hinge Loss kullanılmış ve Eş. 2.2’de tanımlanan enküçültme problemi için eniyilenmiştir. Karşılaştırmalı sonuçlarda önerilen yaklaşımın başarılı olduğu görülmüştür. Lin ve diğerleri mobil görüntü arama işlemlerinde çoklu kod kitabı öğrenimi için çift katmanlı seyrek kodlama önermişlerdir [39]. Shi ve diğerleri SIFT tanımlayıcıları kullanarak SR ile CBIR modeli önermişlerdir [40]. Sift Based Elastic Sparse Coding (SBESC) (ÖBÖD Tabanlı Elastic Seyrek Kodlama – ÖTESK) çalışmasında SR kullanılırken katsayı güncellemesi için hem l_1 -norm hem de l_2 -norm düzenleme terimleri eniyileme probleminde birlikte kullanılmıştır.

$$\arg \min_{\alpha_i} \{ \|x_i - D\alpha_i\|_2^2 + \lambda_1 \|\alpha_i\|_1 + \lambda_2 \|\alpha_i\|_2^2 \} \quad (2.3)$$

Önerilen yaklaşım SIFT karşılaştırma, standart SR + özellik kutusu ve özellik kutusu ile karşılaştırılmış ve %98 kesinlik değeri ile en iyi sonucu vermiştir. Ji ve diğerleri SR ile sıkıştırılmış ve görev bağımlı kod kitapları oluşturmuş ve bunu CBIR dâhil bir takım görüntü işleme probleminde uygulamışlardır [41]. Önerilen yöntemde kod kitapları sınıflama ya da tanıma görevlerine göre özelleştirilir. Burada SR ile kod kitapları oluşturulurken giriş vektörleri terim frekansı ve evrik doküman frekansı prensibine göre ağırlıklandırılarak görsel ayrıştırma sağlanır. Daha sonra görev bağımlı ayrıştırma için etiket-kısıtlı çekirdek kullanılır.

$$\min_{N(D), \alpha_i \in R_k} \{ \| wx_i - D\alpha_i \|_2^2 + \lambda_1 LCK(x_i) + \lambda_2 \| D\alpha_i \|_1 \} \quad (2.4)$$

CBIR işlemlerinde insan yüzü görüntüleri oldukça popülerdir. Chen ve diğerleri SR ile ölçeklenebilir yüz erişimi önermişlerdir [42]. Burada düşük seviye görüntü özellikleri ile yüz özellikleri birleşimi için SR kullanılarak, anlamsal kod kelimelerinin üretimi sağlanmıştır.

$$\min_{\alpha} \sum_{i=1}^n \| x_i - D\alpha_i \|_2^2 + \lambda \| z_i \circ \alpha_i \|_1 \quad (2.5)$$

CBIR işlemlerinde indeksleme ve erişim adımı için özetleme, verimli veri depolama ve hızlı erişim sağlamada iyidir. Zhu ve diğerleri etkili bir özetleme işlemi için SR kullanmışlardır [43]. Çalışmada örnek uzayı, doğrusal spektral kümeleme ile kümelere bölündükten sonra, her bir örneğin ölçeklenmiş olasılıkları seyrek vektörler ile temsil edilir. Elde edilen vektörlerin özet kodlarını elde etmek için en küçük varyans kodlama modeli önerilmiştir. Önerilen yaklaşım yedi özetleme yöntemi ile dört veri kümesi üzerinde karşılaştırılmış ve önerilen yöntem maliyetli olsa da yüksek kesinlik değerleri ile oldukça umut vaat edici sonuçlar vermiştir. Wang ve diğerleri yine insan yüzü erişimi ile ilgili yapmış oldukları çalışmada, bir insan yüzüne sahip görüntüyü benzer insan yüzlerine sahip etiketli görüntüleri kullanarak etiketlemeyi amaçlamışlardır [44]. Çalışmada SR ile yerel tanımlayıcılar öğrenilir ve elde edilen sözlük ile katsayılar kullanılarak etiketli resimlerden etiketsiz resimlerin etiketlenmesi işlemi gerçekleştirilir. Srinivas ve arkadaşları önermiş oldukları Content Based Medical Image Retrieval (CBMIR) (İçerik Tabanlı Medikal Görüntü Erişimi – İTMGE) yaklaşımında hem çevrimdışı hem de çevrimiçi sözlükleri bir arada kullanmışlardır [45]. Algoritmada LFD özellikleri yerine LLF kullanıldığında model oldukça hızlı bir erişim işlemi gerçekleştirmektedir. Zhang ve diğerleri yapmış oldukları çalışmada daha önce literatürde önerilmiş olan bir yaklaşımdaki eksiklikleri gidermek amacıyla iki düzenleme yapmışlardır [48]. Bu düzenlemeler çizge laplas işleci ve negatif olmayan kısıtlardır. Çizge düzenlemesi ile verilerin komşuluk yapısının korunması amaçlanmıştır. Negatif olmayan kısıtlama ise seyrek katsayıların işaretinden etkilenen yerel bilgilerin değerlendirilmesi için adapte edilmiştir.

Geliştirilmiş SR ile yapılan CBIR işlemlerinde görülmüştür ki genellikle veri kümelerinin etiketleri de sözlük ve katsayıların öğrenilmesinde kullanılarak, görüntülerin seyrek temsili oluşturulurken benzer görüntü bölüm ya da tanımlayıcılarının sözlükte bir arada olmaları sağlanmaya çalışılmıştır [30, 37, 38, 41-43, 48]. Böylelikle benzer görüntülerin katsayıları birbirlerine yakın, farklı görüntülerin ise birbirlerinden uzak olması sağlanır. Genel olarak SR ile yapılan CBIR yaklaşımlarında özellik kutusu ya da kelime kutusu yaklaşımları kullanılmaktadır. Özellik kutusu yaklaşımlarında, resmi tanımlayan yerel tanımlayıcılar kullanılır.

Çizelge 2.1. CBIR yaklaşımlarında kullanılan yöntemler

Çalışma	Özellik	Sözlük Öğrenme Algoritması	Katsayı Öğrenme Algoritması
Çalışma [30]	Kıyafet tipi, renk histogramı, Gabor, LBP, SIFT	K-Means	LLC
Çalışma [31]	SURF	Anahtar Kodlama	Anahtar Kodlama
Çalışma [32]	SIFT	K-SVD	SpaRSA
Çalışma [33]	SIFT	K-Means	Homotopy
Çalışma [34]	HOG GIST VE LBP	Çalışma [55]	Herhangi bir eniyileme algoritması
Çalışma [35]	Renk histogramı ve desen momenti	Bütün özellikler	LARS
Çalışma [36]	Harris Detector ve DAISY	Çalışma [56]	Çalışma [56]
Çalışma [37]	Kıyafet tipi, color histogram,Gabor LBP SIFT	K- Means	LLC
Çalışma [38]	Renk, desen, şekil	Çalışma [56]	STMIL
Çalışma [39]	CHoG, MCVD	q-Codebook	Bi Layer SC
Çalışma [40]	SIFT	K-Means	Elastic net
Çalışma [41]	SIFT	İteratif sözlük öğrenme	İteratif sözlük öğrenme
Çalışma [42]	LBP	Rastgele Özellik	LARS
Çalışma [43]	GIST,SIFT	Çalışma [57]	Çalışma [57]
Çalışma [44]	GIST, LBP	Rastgele Özellik	WLRLCC
Çalışma [45]	Piksel yoğunluğunun ortalaması ve varyansı	K-Means ve K-SVD	OMP
Çalışma [46]	IDWT	Tüm özellikler	Çeşitli eniyileme algoritmaları
Çalışma [47]	GALIF	PQ	PQ
Çalışma [48]	SIFT, GIST	NIDL, GNIDL	NOMP, OMP

Çizelge 2.2. SR tabanlı CBIR yaklaşımlarının performansları

Çalışma	Veri Küme(leri)	Performans
Çalışma [30]	Elbise resimleri	Önerilen yöntem en iyi ortalama kesinlik-hatırlama eğrisine sahiptir.
Çalışma [31]	TRECVID 2007	Önerilen yöntem farklı veri boyutları için en iyi Mean Average Precision (MAP) (Ortalama Kesinlik Ortalaması – OKO) değerine sahiptir.
Çalışma [32]	Corel 1000	%100 ortalama kesinlik ilk 10, 20 ve 40 resim için elde edilmiştir.
Çalışma [33]	ImageCLEF VCDT 2008, PASCAL VOC 2010	Her bir açıklama sınıfına göre en iyi Alıcı Çalışma Karakteristiği (AÇK) elde edilmiştir.
Çalışma [34]	Microsoft Research Cambridge veri tabanı	Önerilen yöntem her bir sınıf için en iyi kesinlik-hatırlama eğrisine sahiptir.
Çalışma [35]	Yale face veri kümesi, Corel	Önerilen yöntem her bir veri kümesi için en iyi kesinlik-hatırlama eğrisine ve hesaplama maliyetine sahiptir.
Çalışma [36]	INRIA Holiday, Flickr resimleri, UKB	Önerilen yöntem en iyi MAP değerlerine sahiptir. 0.767 MAP Holiday veri kümesi için, 3.76 erişim skoru UKB için elde edilmiştir.
Çalışma [37]	Elbise resimleri	Önerilen yöntem nesnel değerlendirmelere göre en yüksek memnuniyet skoru elde etmiştir.
Çalışma [38]	Corel, SIVAL	Önerilen yöntemin iki yaklaşımı için en iyi doğruluk elde edilmiştir.
Çalışma [39]	PKUBench, Zubud, UKBench	En iyi MAP ve seyreklik elde edilmiştir.
Çalışma [40]	COIL 20	İlk 20 resim için %98 ortalama kesinlik elde edilmiştir.
Çalışma [41]	UKBench	Erişim işlemi için en iyi sonuç elde edilmiştir.
Çalışma [42]	LFW, Pubfig	%18.6 ve %21 MAP değerleri sırasıyla LFW ve Pubfig için elde edilmiştir.
Çalışma [43]	MNIST, CIFAR, NUS-WIDE, GIST	İki yaklaşım önerilmiştir. SELVE yaklaşımı oldukça maliyetli iken, E-SELVE yaklaşımı en iyi MAP değerini elde etmiştir.
Çalışma [44]	Yüz resimleri, Pubfig	%77 isabet oranı önerilen yöntemin çevrimiçi sürümü için elde edilmiştir. Buna ek olarak çevrimdışı sürümü çalışma zamanını 0.22 saniyeye indirmiştir.
Çalışma [45]	IRMA	%97.14 kesinlik ve %80.1 hatırlama değeri elde edilmiştir.
Çalışma [46]	Flower, Corel, ALOI, VisTex, MPEG-7	Önerilen yöntem en yüksek erişim doğruluğunu elde etmiştir.
Çalışma [47]	Eitz, ETHZ, Flickr15k	En iyi kesinlik-hatırlama eğrisi ETHZ ve Flickr15k veri kümeleri elde edilirken, en iyi Kendall ilgisizlik değeri Eitz veri kümesi için elde edilmiştir.
Çalışma [48]	MNIST, LabelMe, INRIA Holiday, Holidays image and Flickr	Önerilen yaklaşım 4 veri kümesi üzerinde test edilmiş, çizge düzenlemesi daha büyük aram a maliyetine neden olsa da, negatif olmayan kısıtlamalar düzenlemesi sorgu zamanından ödün vermeden daha etkili sonuçlar vermiştir.

Kelime kutusu yaklaşımında ise görüntülerde ortak olarak kullanılan bölgeler ile SR modeli oluşturulur. Özellik çıkarımı stratejisi, farklı sözlük öğrenme algoritmalarının birleşimi ile elde edilen sözlük kullanımı, geliştirilmiş katsayı öğrenme algoritmaları bu çalışmaların özgün yanlarıdır [39, 40, 44-48].

Çizelge 2.1.'de literatürde CBIR için kullanılan SR yaklaşımlarında kullanılan yöntemler gösterilmektedir. Çizelgede her bir çalışma için görüntüden çıkarılan özellikler, katsayı öğrenme algoritması ve kullanılan sözlük için önerilen sözlük öğrenme yaklaşımları açıklanmaktadır. Çizelge incelendiğinde SIFT, SURF, HoG ve LBP yaygın kullanılan özelliklerdir. Özelliklerin basit işlemlerden geçirilerek kullanımı, K-Means (K-Ortalamalar), K- Singular Value Decomposition (K-SVD) (K-Tekil Değer Ayırışma – K-TDA) [50] ve Online Dictionary Learning (ODL) (Çevrimiçi Sözlük Öğrenme – ÇSÖ) [57] sözlük öğrenmesinde etkili çözümler üretir. Orthogonal Matching Pursuit (OMP) (Ortogonal Eşleşme Takibi – OET) [51], homotopy [52], Least-Angle Regression (LARS) (En Küçük Açılı Bağlanım – EKAB) [53] ve elastic net [54] algoritmaları CBIR yaklaşımlarında katsayı öğrenme aşamasında kullanılmıştır.

Çizelge 2.2'de SR tabanlı CBIR yaklaşımlarının performansları özetlenmiştir. Çizelgede kullanılan veri kümeleri ile performans ölçütleri ve bu performans ölçütlerine göre elde ettiği sonuçlar gösterilmiştir. Performans ölçümü bazı çalışmalarda yalnız eğri grafiği olarak verildiği için başarısı ile ilgili net bir ölçüm değeri verilememiştir.

3. ÖZELLİK ÇIKARIMI VE SEÇİMİ

Özellik çıkarımı ve seçimi görüntü işleme yaklaşımlarının başarımında kritik rol oynar. Bu nedenle literatürde pek çok bilgi, görüntünün renk, desen, şekil gibi temel özelliklerini kullanarak görüntüden çıkarılır. Bu bölümde çalışmada kullanılan yerel özelliklerden ve görüntü bölümlenme işlemlerinden bahsedilmiştir.

3.1. Yerel Özellik Tanımlayıcıları

Objeye tanımlama, sınıflandırma ve erişim gibi birçok bilgisayarla görme probleminin çözümü için kullanılan yöntemler temelde görüntüler arasında karşılaştırma yaparlar. Bu karşılaştırma için görüntü içinde bulunan bilgiler çıkarılır ve bu bilgiler özellik olarak adlandırılır. Bilgisayarla görme problemlerinin ilk zamanlardaki çözümlerinde görüntüdeki LLF kullanılır. Bu özellikler görüntüdeki renk, desen, şekil ve mekânsal yerleşim bilgileridir ve küresel özellik olarak adlandırılır [58]. Ancak bu bilgiler görüntü üzerinde yapılan taşıma, döndürme, ölçekleme ve afin işlemlerine karşı dayanıklı değildir [59]. Bununla birlikte görüntüdeki tekrar eden objelerin tanınmasında yetersizdirler. Bu nedenle sonraki yıllarda bu problemlerin çözümü için yerel özellikler önerilmiştir [60]. Yerel özellikler görüntü içindeki yerel yapılarla ilgilenir. Bu yapılar görüntü üzerinde yapılan işlemlere karşı sabittir ve bu nedenle bilgisayarla görme problemleri için tercih edilirler [61]. Yerel özellikler literatürde dağılım tabanlı, mekânsal-frekans ve diğer tanımlayıcılar olarak üç grupta ele alınmıştır [62]. Dağılım tabanlı tanımlayıcılar görüntüdeki bölgelerin ortalama histogramlarını kullanarak görüntüyü temsil ederler. SIFT, SURF, HoG, LBP ve LTP bu grupta yer alan yerel özelliklerdir ve literatürde popüler olarak kullanılmaktadırlar.

3.1.1. SIFT

SIFT özellikleri ilk olarak Lowe tarafından önerilmiştir [14, 63]. Lowe'a göre ölçekleme, döndürme ve ışık değişimine karşı sabit bir özellik iyi bir karşılaştırma bilgisi olarak kullanılabilir. Buna göre önermiş olduğu SIFT özellikleri görüntü ölçekleme ve döndürmeye göre sabit, ışık değişimi ve üç boyutlu kamera görüşüne göre kısmen sabittir [14]. Bununla birlikte bu noktalar mekânsal ve frekans alanında iyi konumlandırılmış ve bozulma olasılığı düşüktür [14]. SIFT özelliklerinin çıkarılması dört temel adımdan

oluşmaktadır. Bu adımlar:

- Ölçek-uzay uç nokta tespiti.
- Anahtar noktaların yerini saptama.
- Yönelim atama.
- Anahtar nokta tanımlama.

Bu adımlara göre ilk olarak görüntü üzerindeki aday anahtar noktalar ölçekleri ile birlikte elde edilir. Bunun için görüntünün farklı ölçeklerine Difference of Gaussian (DoG) (Gauss Farkı - GF) uygulanır. Daha sonra bu ölçekler arasında belirgin noktalar incelenerek aday noktalar tespit edilir. Daha sonra bu aday noktalar içerisinde kullanışlı olmayacağı düşünülen düşük kontrastlı ve kenar bölgelerde olan noktalar elenerek, kalan noktalar anahtar nokta olarak belirlenir. Belirlenen anahtar noktalara yönelim atama işlemi gerçekleştirilir. Bu işlem görüntüdeki yerel eğim yönlerine göre belirlenir. Bazı durumlarda bir noktaya birden fazla yönelim atanabilir. Görüntüden elde edilen bu anahtar noktalar şuan için ölçek ve döndürme işlemlerine sabittir. Son olarak anahtar nokta tanımlama işlemi gerçekleştirilir. Bu adımda her bir anahtar nokta için merkezi kendisi kabul edilen ve bulunduğu ölçek ile orantılı bir alan içindeki piksellerin eğim büyüklükleri ve yönelim açısı belirlenir. Böylece anahtar nokta ışık değişimine karşı da sabit olacaktır. Sonuç olarak bu adımlar sonunda elde edilen her bir SIFT anahtar noktası ölçek, dönüşüm ve ışık değişimine karşı sabit olacaktır.

3.1.2. SURF

Yerel özelliklerin verimli kullanımı ile birlikte bu alandaki çalışmalar da önemli bir konu haline gelmiş ve yerel özelliklerin tespit ve tanımlama performansını artırma amaçlı çeşitli çalışmalar yapılmıştır. Bu noktada SURF, SIFT özelliklerine alternatif olması amacı ile ortaya çıkmıştır ve bu nedenle SIFT algoritmasının iyileştirilmiş hali olarak düşünülebilir [59].

SURF özellikleri ilk olarak Bay ve diğerleri tarafından önerilmiştir [15, 64]. Bay ve diğerlerine göre bir anahtar noktanın en önemli özelliği tekrarlanabilir olmasıdır [15]. Böylece bu anahtar nokta, farklı görüşlerden bakıldığında benzer bir davranış sergileyecek

ve güvenilir olacaktır.

Tespit ve tanımlama olarak iki temel adımdan oluşur. SURF hızlı bir şekilde anahtar noktaların tespitini yapabilmek için ilk adım olarak integral görüntüleri kullanan Hessian matris kullanır. Bu işlem algoritmanın performansını arttıran en önemli adımdır. Görüntünün ölçeklemesi ve bu ölçeklere göre anahtar noktaların konumlandırılması ile birlikte tespit adımı sona erer ve tanımlama adımına geçilir. Bu adımda yine SIFT yaklaşımında olduğu gibi yönelim atama işlemi gerçekleştirilir. Burada Haar dalgacıkları kullanılarak her bir anahtar nokta için çevresindeki piksellerin yönelimi belirlenir.

3.1.3. HoG

HoG ilk olarak Dalal ve Triggs tarafından insan tespitinde kullanılmıştır [16]. Bu yöntem yerel eğimlerin yönlerinin hesaplanması prensibine dayandığından SIFT yönteminin basitleştirilmiş bir çeşidi olarak düşünülebilir. Algoritmanın adımları şu şekildedir:

- Görüntünün renk ve gama değerlerini normalleştirme.
- Görüntünün hücrelere bölünerek her bir hücreye ait eğimlerini hesaplama.
- Bu hücreler üzerinde oluşturulan blokların her birinin histogramlarını normalleştirme.
- Blokların artarda eklenerek HoG özelliklerinin elde edilmesi.

3.1.4. LBP ve LTP

Görüntülerin analizinde desen önemli bir küresel özelliktir. LBP yaklaşımı yerel desenleri ayrıştırma konusunda oldukça başarılı bir tekniktir [17, 65]. Wang ve He tarafından önerilen desen analizi yönteminin iki seviye sürümü olarak Ojala ve diğerleri tarafından geliştirilmiştir [65]. Bu yönteme göre bir desen görüntüsü, desen birimi olarak adlandırılan alt desen parçaları ile ifade edilebilir. Bu birimler sekiz elemanlıdır ve bu elemanlar o birimin komşularına göre belirlenmektedir. Bir başka deyişle LBP özelliği görüntüdeki piksellerin komşu piksel değerleri ile arasındaki karşılaştırmaya göre belirlenir. Bir görüntü üzerinde LBP özellikleri çıkarılırken aşağıdaki adımlar izlenir:

- Görüntü üzerindeki her bir piksel için 3x3'lük bir pencere içerisindeki komşuları ile karşılaştırma yapılır. Burada pencere boyutu farklı olabilir. Buna göre pikselden küçük olan komşular 0, büyük veya bu piksele eşit olanlar ise 1 olarak işaretlenir ve sekiz elemanlı bir birim elde edilir. Bu birim onluk sayı sistemine çevrilerek pikselin yeni değeri belirlenmiş olur.
- Tüm piksellerin yeni değerleri belirlendikten sonra görüntü bloklara bölünür. Her bir blok için bu piksel değerlerine göre histogramlar oluşturulur. Burada pikseller 256 farklı değer alacağı için histogram boyutu 256 olur. Ancak burada piksellerin değerlerinin ikilik sayı sistemindeki haline göre düzenli olanları alınır. Sekiz bitlik bir seride 58 adet düzenli LBP kodu vardır. Düzenli olmayan LBP kodları için tek bir sütun histogram üzerinde yeterli olacağından histogramın boyutu 59'a indirgenmiş olur. Böylece her bir blok için 59 boyutlu bir vektör elde edilmiş olur ve tüm görünüş için LBP özellikleri elde edilir. Burada bloklar artarda birleştirilerek tek bir özellik olarak düşünülebileceği gibi, her bir blok bir özellik olarak da düşünülebilir [22].

LTP yaklaşımı ise LBP yaklaşımın geliştirilmiş bir sürümüdür [18, 66]. Burada özelliklerin çıkarılması işlemi LBP ile aynıdır. Ancak burada her bir piksel üst ve alt olarak iki farklı değer alır. Bu değer karşılaştırma işlemi yapılırken bu pikselin değerine bir eşik değeri eklenmesi ile oluşturulur. Buna göre pikselin değeri ile eşik değeri toplamından küçük olan değerler 0, büyük veya bu değere eşit olan değerler 1 olarak işaretlenir ve sekiz elemanlı birim bu pikselin üst değeri olarak elde edilir. Pikselin değeri ile eşik değeri farkından küçük olan değerler 0, büyük veya bu değere eşit olan değerler 1 olarak işaretlenir ve sekiz elemanlı birim bu pikselin alt değeri olarak elde edilir. Sonraki adımda ise yine LBP özellik çıkarımında olduğu gibi görüntü bloklara ayrılır ve bu sefer piksellerin üst ve alt değerleri için 59 boyutlu histogramları oluşturulur ve böylece LTP özellikleri çıkarılmış olur. Buna göre aynı komşuluk çerçevesi ve blok sayısı için LTP özellikleri, LBP özelliklerinin iki katı olacaktır.

3.2. Görüntü Bölümleme

Bir görüntüyü bir arada anlamlı bilgiler içeren özelliklere ayırma işlemi görüntü bölümleme olarak adlandırılır. Görüntü bölümleme yüksek seviye anlamsal bilgilerin elde edilmesinde etkili çözümler üretir. Özelliklerin bir arada sergilediği anlamlı bilginin kullanılmasını sağlamanın yanı sıra görüntü üzerinde ilgisiz bölgelerin çıkarılmasını da

sağlar. Bu nedenle birçok bilgisayarla görme probleminde önemli bir ön işleme adımı olarak kullanılır. Literatürde birçok görüntü bölümlenme algoritması önerilmiş olduğu için bu algoritmaların gruplanması ihtiyacı oluşmuştur. Literatürde birçok tarama çalışması yapılmış ve bölümlenme algoritmaları gruplandırılmıştır [67, 68]. Literatürdeki görüntü bölümlenme algoritmalarını bölge-tabanlı, kenar-tabanlı, çizge-tabanlı, şekil-tabanlı, eşik-tabanlı ve kümeleme-tabanlı bölümlenme olarak ayırabiliriz. Bu yaklaşımlara ek olarak, tüm veri tabanı görüntüleri kullanılarak yapılan birlikte- bölümlenme yaklaşımları da literatürde yeni çalışılan bir alandır [67]. Buna göre tüm görüntülerdeki ortak bölgeler çıkarılır ve veri tabanı ile ilgili anlamlı bir çıkarım yapılır. Bu yaklaşım sınıflandırma, erişim ve otomatik ek açıklama gibi bilgisayarla görme problemlerinde tek bir görüntü üzerinde yapılan bölümlenme işlemine göre daha etkili sonuçlar vermektedir.

Bölgesel-tabanlı bölümlenme yaklaşımlarında aynı bölgede olan özellikler benzer değerlere sahiptirler. Bu nedenle her bir özelliğin komşularına bakılarak bölümlenme işlemi yapılır. Genel yaklaşımda, her bir pikselin komşuları ile olan benzerliğine bakılır, eğer kıstas sağlanıyorsa bu pikseller aynı kümeye atanır. Bölge genişlemesi, böl ve birleştir olarak iki yaklaşımı literatürde sıklıkla görülmektedir [68]. Tohumlu Bölge Genişlemesi algoritması literatürdeki en basit bölgesel-tabanlı bölümlenme yaklaşımıdır. Bölge genişlemesi yaklaşımı olan Tohumlu Bölge Genişlemesi algoritmasının Tohumsuz Bölge Genişlemesi türevi de literatürde kullanılmaktadır. Tohumlu Bölge Genişlemesi algoritmasında görüntü üzerinde belirli C adet küme noktası belirlenir. Her bir küme noktası komşularıyla kıyaslanır ve eğer benzerlik koşulu sağlanıyorsa bu pikseller bu küme noktasının ait olduğu kümeye atanır. Daha sonra bu küme noktasının sınır noktaları yeni tohum noktası olarak belirlenir ve küme noktasının merkezi yeniden hesaplanır. Böl ve birleştir yaklaşımında ise görüntü homojen bölgelere ayrılır. Ağaç tabanlı bir yaklaşım olup her bir düğüm dört çocuk düğüme sahiptir. Bu yaklaşımlarda ilk olarak tüm görüntü tek bir düğüm olarak ele alınır ve ayırma işlemi başlatılır. Eğer bir bölge uygunluk ölçütünü sağlamıyorsa dört alt bölgeye ayrılır. Bu işlem ayrılabilen alan kalmayana kadar devam eder ve görüntü bir ağaç yapısında bölgelere ayrılır. Daha sonra birleştirme işlemi yapılır ve burada birbirine benzer bölgeler birleştirilir. Böylece algoritma tamamlandığında görüntü belirli sayıda bölgelere ayrılmış olur. Burada ayırma ve birleştirme ölçütleri farklı olur.

Kenar-tabanlı bölümlenme bölgesel tabanlı bölümlenme aksine bu yaklaşımda, öncelikle objeler etrafındaki kenarlar belirlenir ve bu kenarlar objenin sınırlarını oluşturduğunda içleri doldurularak bölümlenme işlemi gerçekleştirilir [68]. En basit hali ile ilk olarak tüm görüntü üzerindeki kenarlar hesaplanır. Daha sonra bu kenarlar üzerinde işlem yapılarak yalnızca objeleri sınırlayan kenarların kalması sağlanır. Son olarak sınırların içi doldurularak objeler elde edilir ve bölümlenme işlemi tamamlanmış olur. Elde edilen kenarlardan yalnız obje sınırlarının kalması en zor adımdır. Gürültü ve diğer etkenler ilgili kenarların belirlenmesinde karşılaşılan zorluklardır. Bununla birlikte objelerin etrafında olması gereken kenarlar tespit edilmediğinde, obje etrafındaki diğer kenarlar arasında oluşan boşlukları kapatacak kenar bağlama olarak adlandırılan akıllı yaklaşımlara da ihtiyaç vardır.

Çizge-tabanlı algoritmalarda görüntü pikselleri köşe, pikseller arasındaki benzerlik ise kenar ağırlığı olarak belirlenir. Burada amaç görüntüyü ayrık çizge kümesi olarak göstermektir. Her bir alt çizge görüntüdeki bir bölgeyi temsil etmektedir. Normalleştirilmiş kesim en popüler çizge tabanlı bölümlenme algoritmasıdır [69].

Şekil-tabanlı bölümlenme yaklaşımları görüntüdeki şekiller etrafındaki sınırları belirlemeyi amaçlar [7]. Aktif Şekil Model (AŞM) ve Aktif Görünüm Model (AGM) bu kategorideki en popüler iki bölümlenme algoritmasıdır. AŞM algoritmasında bir eğitim seti olarak belirlenen bir şekil görüntü içerisinde aranır.

Eşik-tabanlı bölümlenme yaklaşımları genellikle histogram eşikleme yaparak görüntüyü bölütler. En basit şekliyle gri seviye bir görüntüyü iki farklı kümeye ayırmak için bir eşik kullanılır. Bu eşik değerinden küçük piksellere 0 büyük ya da eşit olan piksellere 1 atandığında bölümlenme işlemi gerçekleştirilmiş olur. Siyah arka plan üzerindeki parlak objeleri ayırmak için tek bir eşik değeri yeterli olabilir. Ancak bazı durumlarda daha fazla eşik değerine ihtiyaç duyulur. Bu durumlarda görüntüde kullanılacak eşik değerini belirlemek için genellikle tüm görüntüden elde edilen histogram grafiğine bakılır. Daha sonra piksellerin yoğunlaştığı bölgelerde ayırma yapılacak şekilde eşik değerleri belirlenir.

Kümeleme algoritmaları bölümlenme işlemlerinde sıklıkla kullanılmaktadır. Kümeleme algoritmaları veri kümesindeki elemanların belirlenen merkezlerle temsil edilmesidir. Burada benzer veriler aynı merkez ile temsil edilir ve böylece kümeleme işlemi yapılır.

Hiyerarşik ve ayırma tabanlı iki kategoride deęerlendirilir. Hiyerarşik kümeleme algoritmaları veri kümesini bir ağaç yapısında iç içe benzerlik seviyesine göre gruplar. Burada verilerin her biri ilk başta ayrı merkez deęeri olarak deęerlendirilip en benzer olanları birleřtirilebilir ya da belirli sayıda belirlenen merkez deęerleri verileri temsil ederek ayırma işlemleri yapılabilir. Ayırma-tabanlı kümeleme algoritmaları hiyerarşik yaklaşımların aksine başlangıçta tek bir bölüm kullanarak kümeleme işlemine başlar. K-Means algoritması bu kategorideki en popüler algoritmadır. Bu algoritmaların en büyük dezavantajı küme sayısının bilinmesi ihtiyacıdır. Ancak büyük veri kümelerinde hiyerarşik kümeleme algoritmalarının aksine daha hızlıdır. Mean Shift (Ortalama Kayma) algoritması da yoğunluk tahmini tabanlı bir dięer popüler kümeleme-tabanlı bölümlenme algoritmasıdır [70].

Literatürde CBIR yaklaşımları incelendiğinde K-Means ve Mean Shift algoritması popüler olarak kullanılmıştır [71-74]. Bu çalışmada bu algoritmaların görüntüdeki piksel deęerlerini kullanarak yaptığı bölümlenme yaklaşımının yanı sıra, piksel deęerleri ve onların konum bilgilerini de kullanarak yaptığı bölümlenme işlemleri incelenecektir.



4. SEYREK TEMSİL

SR istatistik, makine öğrenmesi ve görüntü işleme gibi birçok disiplinde kullanılan bir tekniktir ve seyreklik (sparsity) prensibine dayanır [21]. Birçok alanda, eldeki problemin çözümü için geliştirilecek olan modellerden en basit olanının seçilmesi oldukça eski bir yaklaşımdır. Burada problemin çözümü için izlenen yol şu şekildedir, oluşturulan modeller gözlemlenen verinin eğitim için ayrılmış kısmı kullanılarak oluşturulduktan sonra, bu modellerin tahmin gücü gözlemlenen verinin test için ayrılan kısmı ile sınılanır. Burada araştırmacılar için model seçimi yapılırken dikkat edilecek en önemli hususlardan birisi de model içinde kullanılan parametrelerin sayısıdır. Çünkü bu kıstas modelin performansını belirler. Seyreklik prensibi vektör olarak ifade edilen herhangi bir verinin belirli bir tabana göre daha basit katsayılarla ifade edilmesidir [20]. Böylece model oluşturulurken birçok parametre elenerek daha az sayıda parametre ile model basitleştirilir. Lineer cebir tarafından inceleyecek olursak, herhangi bir vektör kümesinin bir taban üzerindeki ifadesi bu vektör kümesinin bu taban üzerinde konumunu verir. Böylelikle vektör kümesindeki elemanların birbirleri ile mekânsal bir ifadesi elde edilmiş olur. Seyreklik prensibi daha önce de bahsedildiği gibi birçok alanda karşımıza çıkmaktadır. Bu çalışmanın konusu olan görüntü için değerlendirdiğimizde, seyreklik prensibi bir görüntünün özelliklerinin belirli bir sözlüğün elemanları ile basit doğrusal birleşimleri ile yeniden ifade edilmesidir. Buna göre problem aşağıdakilerden herhangi birisi olabilir.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|x - D\alpha\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_p \leq k \quad (4.1)$$

$$\min_{\alpha \in \mathbb{R}^n} \|\alpha\|_p \quad \text{s.t.} \quad \|x - D\alpha\|_2^2 \leq \varepsilon \quad (4.2)$$

Burada x özniteliği için D sözlüğü üzerinde α katsayısı ile ifade edilmeye çalışırken iki farklı yaklaşım kullanılabilir. Eş. 4.1'de seyreklik k adet sayı ile sınırlandırılırken, Eş. 4.2'de ise özellik ve doğrusal birleşimi arasındaki fark bir üst hata sınırı ile sağlanmaya çalışılır. Bu denklemlerin Lagrange ifadesi Eş. 4.3'tedir.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_p \quad (4.3)$$

Burada seyreklik teriminin enküçüklemesi genel bir ifade ile p -norm olarak ifade edilmiştir ancak burada $p=[0-\infty]$ değerler alabilmektedir. Bu enküçükleme l_0 -norm için NP-Hard problemidir [75]. Bu nedenle literatürde l_1 -norm (Lasso) ya da l_2 -norm popüler olarak kullanılır. Bu iki enküçükleme temel SR yaklaşımlarında uygulanır. Ancak çekirdek tabanlı enküçükleme yaklaşımları da literatürde mevcuttur [76]. Buna göre Eş. 4.3'ün en genel hali aşağıdaki gibi ifade edilebilir.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|x - D\alpha\|_2^2 + J(\alpha) \quad (4.4)$$

Burada $J(\alpha)$ seyrekliği elde etmemiz için gerekli maliyet fonksiyonumuz olarak tanımlanır. SR oluşturmak için iki adım vardır. Birinci adım sözlüğün öğrenilmesi, ikinci adım ise seyrek katsayıların oluşturulmasıdır. Literatürde bu adımlar için çeşitli algoritmalar önerilmiştir. Katsayıların oluşturulması için kullanılan algoritmalar Eş. 4.4'ü ele alış biçimlerine göre değişmektedir. SR bilgisayarla görme işlemlerinde yoğun olarak kullanılmaktadır. Görüntü gürültü temizleme, görüntü iç boyama, görüntü sıkıştırma, görüntü büyütme, görüntü sınıflandırma ve görüntü erişimi bu problemlerden bazılarıdır [20, 75]. SR bu problemlere ürettiği çözümlerde farklı çatılar kullanır. Görüntü erişimi ve görüntü sınıflandırma problemleri için SR ile oluşturulacak çatı şu şekilde özetlenebilir.

- Genellikle kullanılan sözlük birden fazla görüntünün temsil edildiği bir taban görevini görür [20]. İkili karşılaştırma durumlarında dahi en az iki resim için bir sözlük oluşturma gereksinimi vardır [32].
- Her görüntü için birden fazla özellik vektörünün seyrek katsayıları bu sözlük üzerinde katsayı öğrenme algoritması kullanılarak oluşturulur.
- Her bir görüntünün özellikleri için elde edilen katsayılar havuzlama ile birleştirilerek karşılaştırma işleminde kullanılacak tek bir katsayı vektörü elde edilir. Bu vektörün birçok seyrek katsayının havuzlanmasından elde edilmesinden dolayı seyrek olması beklenmez. Havuzlama işlemi ortalama ya da en büyük gibi küresel havuzlama ya da mekânsal havuzlama gibi yerel havuzlama olarak farklı şekillerde yapılabilir [36, 77, 78].
- Havuzlama yapılarak tek bir görüntü için elde edilen vektör, diğer görüntülerle karşılaştırılarak süreç tamamlanır.

SR yaklaşımının iki temel adımı detaylı olarak incelenecektir.

4.1. Sözlük Öğrenme

Sözlük öğrenmesi SR yapılırken kullanılan maliyet fonksiyonlarındaki en önemli bileşendir. Vektör Quantization (VQ) (Vektör Nicemleme - VN) bilgisayarla görme problemlerinde oldukça popüler olarak kullanılmış ve başarılı sonuçlar vermiştir. Ancak sonraki yıllarda SR kullanımının bu yönteme kıyasla daha başarılı olduğu gözlemlenmiştir [79]. Sözlüğün performans artışıdaki etkisini görmek amacı ile Coates ve Ng ayrıntılı bir araştırma yapmıştır [79]. Yine benzer bir çalışma Rubinstein ve diğerleri tarafından gerçekleştirilmiştir [80]. Burada görüntü problemleri için kullanılan dönüşüm yöntemlerinden modern sözlüklere kadar geniş bir zaman aralığında yapılan çalışmalar incelenmiştir. Literatürde SR için kullanılan sözlükler iki grupta incelenir. Birincisi çevrimdışı olarak, katsayılar oluşturulmadan önce oluşturulan sözlüklerdir. VQ olarak K-means ve var olan özelliklerin normalleştirilerek birleştirilmesi yaklaşımı en popüler olan iki yaklaşımdır [50]. İkinci grupta ise veri tabanındaki her bir görüntü için katsayılar oluşturulurken sözlüğün de bu sırada eğitilmesi işlemidir. Bu grupta Methods of Optimal Directions (MOD) (En uygun Yönler Yöntemi - EYY) ve K-SVD algoritmaları popüler olarak kullanılmaktadır [80]. Bu yaklaşımlarda görüntü veri tabanı için SR oluşturulurken, öncelikle sözlük var olan rastgele katsayılarla göre eğitilir. Daha sonra bu eğitilmiş sözlüğe göre katsayılar eğitilir. Bu işlem belirli bir sonlandırma ölçütü gerçekleşinceye kadar sırası ile tekrar eder.

SR için etkili bir sözlük öğrenme algoritması geliştirmek amacıyla Aharon ve diğerleri K-SVD algoritmasını önermişlerdir [50]. Burada amaç verilen bir eğitim kümesindeki her bir eleman için en seyrek katsayıları üreten sözlüğün elde edilmesidir. K-Means algoritmasının genelleştirilmiş hali olan K-SVD algoritması MOD algoritması yaklaşımında olduğu gibi iç içe enküçükleme tekniği kullanan tekrarlayan bir yöntemdir. Algoritmanın en temel adımı sözlükteki her bir atomun SVD ile eğitilmesidir. Bu işlem K kez gerçekleştirilir ki bu sözlükteki atomların sayısıdır. K-SVD algoritmasının en büyük avantajı sözlük atomlarının eğitiminde matris tersi kullanmadan her bir atomu tek tek basit bir şekilde güncellemesidir. Bu yaklaşım MOD algoritması ile kıyaslandığında büyük bir performans artışı vermektedir [50]. Bir sözlük öğrenme algoritmasının belirli özellikleri sağlaması gerekmektedir. Bu özelliklerden ilki algoritmanın esnekliğidir. Buna göre

algoritma katsayı öğrenme aşamasında herhangi bir katsayı öğrenme algoritmasını kullanabilmelidir. İkinci özelliği basitliğidir. Her ne kadar MOD algoritması bu özelliği taşımaya yakın olsa da halen geliştirilmeye ihtiyacı vardır. Üçüncü özelliği verimli olmasıdır. Yani algoritma hızlı bir şekilde yakınsayabilmelidir. Son olarak ise iyi tanımlanmış ve nesnel olmalıdır. Bir başka deyişle yöntemin başarısı elde edilen sonucun kalitesini iyi ölçen bir nesnel fonksiyon ile gerçekleştirilir. Bu özelliklere sahip olmayı amaçlayan K-SVD algoritmasının adımları şu şekildedir.

- Rastgele bir D sözlüğü oluştur ve sütunlarını normalleştir.
- α katsayılarını elde etmek için herhangi bir katsayı öğrenme algoritmasını kullan.
- Sözlükteki her bir sütun için, bu sütunu kullanan katsayıları grupta ve her bir sütun için toplam temsil hatasını hesapla:

$$E_k = X - \sum_{j \neq k} d_j \alpha_T^j \quad (4.5)$$

- SVD ayrıştırmayı uygula.
- Eğer $\|X - D\alpha\|_2^2$ içindeki değişiklik yeterince küçük ise algoritmayı sonlandır. Değilse katsayı öğrenme adımına dön.

Burada bahsedilen dört yaklaşım dışında da literatürde önerilmiş olan sözlük öğrenme algoritmaları mevcuttur [56, 57, 81].

4.2. Katsayı Öğrenme

SR yaklaşımları için ikinci kritik adım katsayıların öğrenilmesidir. K-Means tabanlı sözlük öğrenme algoritmalarının aksine bu adım için önerilen yaklaşımlar oldukça fazladır [82]. Bu çalışmada bu yaklaşımlardan CBIR için yoğun olarak önerilmiş olan beş katsayı öğrenme algoritması incelenmiştir. Buna ek olarak yakın zamanda önerilmiş olan ve CBIR problemlerinde kullanılmamış olan iki tekrarlayan-daralma algoritması da bu çalışmada incelenmiştir. Bu algoritmaların CBIR yaklaşımlarındaki performansı da önerilen CBIR çatısı ile test edilecektir.

4.2.1. OMP

Literatürde en yoğun olarak kullanılan algoritma OMP algoritmasıdır [51]. Katsayı vektörünün o anki değeri için her seferinde bir girişini seçerek güncelleme işlemi yapan, tekrarlayan bir algoritmadır. Ancak bu işlemi yaparken kalan kısmın, seçilen girişlere dikey olmasını sağlar ki bu standart Eşleşme Takibi algoritması ile arasındaki temel farktır. Böylece algoritma tek bir seferde katsayı içindeki tüm sıfır olmayan değerleri günceller. OMP algoritması Eş. 4.6'yı enküçükleyerek en uygun seyrek katsayıyı bulmayı amaçlar.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|x - D\alpha\|_2^2 \quad \text{s.t.} \quad \|\alpha\|_0 \leq k \quad (4.6)$$

Burada l_0 -norm enküçüklemesi yapılırsa da literatürde stOMP gibi Eşleşme Takibi tabanlı tekrarlayan-daralma metotları gibi farklı OMP yaklaşımları da mevcuttur [83]. OMP algoritması Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. OMP algoritması

D : Sözlük

x : Görüntü özelliği

k : İterasyon indeksi; $k = 0$

α^0 : Başlangıç çözümü; $\alpha^0 = 0$

r^0 : Başlangıç kalan değeri; $r^0 = x - D\alpha^0$

S^0 : Başlangıç giriş indeksi çözüm kümesi; $S^0 = \{\emptyset\}$

ε^0 : Beklenen hata;

1. $k = k + 1$

2. $\varepsilon(j) = \min_{z_j} \|d_j z_j - r^{k-1}\|_2^2$ hatasını her bir sözlük atomu j için en uygun $z_j^* = d_j^T r^{k-1} / \|d_j\|_2^2$ ile hesapla

3. Minimum hatayı veren atom j^* çözüm kümesine ekle
 $S^k = S^{k-1} \cup \{j^*\}$

4. Çözüm kümesine göre o anki katsayı α^k 'yi hesapla.

5. Kalanı hesapla $r^k = x - D\alpha^k$

6. Eğer $\|r^k\|_2 < \varepsilon_0$ algoritmayı durdur. Değilse Adım 1'e dön.

4.2.2. Homotopy

Homotopy algoritması OMP algoritmasında olduğu gibi, boş bir çözüm kümesi ile başlar ve tekrarlayan olarak çözüm kümesini günceller [52]. Ancak buradaki işlem l_1 -norm için gerçekleştirilir. Homotopy algoritmasının en küçüklediği fonksiyon Eş. 4.3'ün l_1 -norm formudur. Burada sonlandırma ölçütü λ parametresinin tüm değerleri içinde en ideal çözüm olabileceği gibi, bir hata üst sınırı ya da belirli sayıdaki sıfır olmayan eleman için de gerçekleştirilebilir. Algoritmanın adımları Çizelge 4.2'de gösterilmektedir.

Çizelge 4.2. Homotopy algoritması

<p>D : Sözlük x : Görüntü özelliği λ : Düzenleme Katsayısı; $\lambda = \ D^T x\ _\infty$ α : Başlangıç çözümü; $\alpha = 0$ Γ : $\{j \in \{1, \dots, p\} : d_j^T x = \lambda\}$</p> <ol style="list-style-type: none"> $\lambda > 0$ olduğu sürece aşağıdaki adımları yap. Yoksa sonlandır. Düzenleme yolunun o anki yönünü hesapla. $\alpha_\Gamma^* = (D_\Gamma^T D_\Gamma)^{-1} (D_\Gamma^T x - \lambda \eta_\Gamma)$ $\alpha_{\Gamma^c}^*(\lambda) = 0$ Aşağıdaki koşulları sağlayan en küçük $\tau > 0$ değerini bul. <ol style="list-style-type: none"> Eğer $d_j^T (x - D\alpha^*(\lambda - \tau)) = \lambda - \tau$ koşulunu sağlayan bir j varsa $\Gamma = \Gamma \cup \{j\}$ Eğer $\alpha^*(\lambda - \tau)[j]$ sıfıra geliyorsa $\Gamma = \Gamma - \{j\}$ $\lambda = \lambda - \tau$; Adım 1'e dön.

4.2.3. LARS

Eş. 4.3'te tanımlanan fonksiyonun l_1 -norm formu makine öğrenmesinde Lasso olarak adlandırılır ve regresyon amaçlı kullanılır. Burada verilen bir giriş için, sözlük üzerinde çok az sayıda sıfır olmayan değere sahip bir katsayı ile doğrusal bir birleşim oluşturmak, model seçimi olarak adlandırılır [53]. LARS algoritması da Homotopy algoritmasında olduğu gibi tüm olası λ değerleri için çözüm kümesini arar. Benzer şekilde OMP algoritmasında olduğu gibi yinelemeli olarak her seferinde çözüm kümesine sözlükteki bir

atomu ekler. Bu nedenle Homotopy algoritmasının geliştirilmiş sürümü olarak düşünülebilir.

4.2.4. Elastic net

Elasticnet Zou ve Hestie tarafından önerilmiştir [54]. Algoritmanın en belirgin özelliği en küçükleme fonksiyonunda l_1 ve l_2 normu birlikte kullanmasıdır.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda_1 \|\alpha\|_1 + \lambda_2 \|\alpha\|_2 \quad (4.7)$$

Algoritma yeni bir düzenleme yaklaşımı ve özellik seçimi amacıyla önerilmiştir. Burada Lasso yaklaşımındaki sınırlamaları gidermek amacıyla l_2 -norm da enküçüklemeye dâhil edilmiştir. Yazarlara göre Lasso algoritmasının bir takım sınırlamaları vardır. Bunlar gözlem sayısının gözlem vektör boyutundan çok az olduğu durumlarda; ($p > n$) l_1 -norm kullanımından dolayı en fazla n adet atomun seçilmesi, bir grup içinde birbirleri ile yüksek ilişime sahip değişkenler olduğunda algoritmanın bunlardan yalnızca birini seçmesi ve son olarak gözlem sayısının gözlem vektör boyutundan büyük olduğu durumlarda ($n > p$) veya belirleyiciler arasında yüksek ilinti olduğu durumlarda, Lasso enküçüklemesinin Ridge Regression (RR) (Çizgi Bağlanımı - ÇB) tarafından denetim altına alınmasıdır [54].

4.2.5. Tekrarlayan daralma algoritmaları

Yapılan çalışmalar göstermiştir ki seyrek çözümlerde Homotopy ve OMP gibi açgözlü yaklaşımlar yüksek boyutu problemler için uygun değildir [84]. Yakın zamanda yapılmış olan çalışmalardan tekrarlayan-daralma yaklaşımları yüksek boyutlu problemlerde etkili ve verimli çözüm üretebilmektedir. Elad bu yaklaşımlarla ilgili yapmış olduğu çalışmasında Seperable Surrogate Functionals (SSF) (Ayrılabilir Temsil Fonksiyonları – ATF) [85] ve Parallel Coordinate Descent (PCD) (Paralel Koordinat Düşümü – PKD) [86, 87] yaklaşımlarının etkili olduklarını belirtmektedir [84]. Bununla birlikte bu yaklaşımları hızlandırmak için Sequential Subspace Optimization (SESOP) (Sıralı Altuzaylık Optimizasyonu – SAO) [88] yaklaşımlarının oldukça verimli olduğunu yapılan çalışmada kanıtlanmıştır [84].

SSF algoritması Daubechies ve diğerleri tarafından önerilmiştir [85]. Algoritmada SR için kullanılan enküçükleme fonksiyonu yerine bir vekil fonksiyon kullanılmakta ve enküçükleme işlemi bu şekilde gerçekleştirilmektedir. Enküçükleme fonksiyonu ve vekil fonksiyon elde etmek için kullanılan ek terim sırasıyla Eş. 4.8 ve Eş. 4.9'da tanımlanmıştır.

$$f(\alpha) = \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda 1^T p(\alpha) \quad (4.8)$$

$$d(\alpha, \alpha_0) = \frac{c}{2} \|\alpha - \alpha_0\|_2^2 - \frac{1}{2} \|D\alpha - D\alpha_0\|_2^2 \quad (4.9)$$

Buna göre Eş. 4.8 yerine kullanılacak vekil fonksiyon Eş. 4.10'da tanımlanmıştır.

$$f^*(\alpha) = \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda 1^T p(\alpha) + \frac{c}{2} \|\alpha - \alpha_0\|_2^2 - \frac{1}{2} \|D\alpha - D\alpha_0\|_2^2 \quad (4.10)$$

Algoritmanın adımları Çizelge 4.3'te verilmiştir.

Çizelge 4.3. SSF-SESOP

<p>D : Sözlük x : Görüntü özelliği k : İterasyon indeksi; $k = 0$ α^0 : Başlangıç çözümü; $\alpha^0 = 0$ r^0 : Başlangıç kalan değeri; $r^0 = x - D\alpha^0$ e : Hata; t : Sonlandırma eşik değeri;</p> <ol style="list-style-type: none"> 1. $k = k + 1$ 2. $e = D^T r_{k-1}$ 3. Daraltma adımı: $e_s = Shrink(\alpha_{k-1} + e / c)$ 4. Maliyet fonksiyonunu minimize edecek μ değerini seç 5. $\alpha_k = \alpha_{k-1} + \mu(e_s - \alpha_{k-1})$ 6. $r^k = x - D\alpha^k$ 7. Eğer $\ \alpha_k - \alpha_{k-1}\ _2 < t$ algoritmayı durdur. Değilse Adım 1'e dön.

Bir diğer tekrarlayan-daralma algoritması Elad ve diğerleri tarafından önerilmiş olan PCD algoritmasıdır [86, 87]. Koordinat düşüm bir başka deyişle Eşleşme Takibi algoritmaları

OMP algoritması anlatılırken bahsedildiği gibi her yinelemede bir atomu seçerek katsayıları eniyiler. PCD ile bu işlem paralel olarak gerçekleştirilir. İşlemin paralel olarak gerçekleştirilmesi seyrek katsayı elde etme süresini oldukça kısaltır. Böylelikle büyük veri kümelerinde oldukça etkili sonuçlar üretir. Tekrarlayan-daralma algoritmaları görüntü erişim işlemlerinde şu ana kadar kullanılmamıştır. Ancak diğer bilgisayarla görme problemlerinde ürettiği sonuçlardan dolayı bu çalışmada bu algoritma da incelenmiştir.

Algoritmanın adımları Çizelge 4.4'te gösterilmektedir. Çizelge 4.3 ile karşılaştırıldığında temel farklılığın daraltma adımında olduğu gözlemlenmektedir. Bu adım dışında algoritma adımları benzerdir.

Çizelge 4.4. PCD-SESOP

<p>D : Sözlük x : Görüntü özelliği k : İterasyon indeksi; $k = 0$ α^0 : Başlangıç çözümü; $\alpha^0 = 0$ r^0 : Başlangıç kalan değeri; $r^0 = x - D\alpha^0$ X : Ağırlık matrisi; $X = \text{diag}(D^T D)^{-1}$ e : Hata; t : Sonlandırma eşik değeri;</p> <ol style="list-style-type: none"> 1. $k = k + 1$ 2. $e = D^T r_{k-1}$ 3. Daraltma adımı: $e_s = \text{Shrink}(\alpha_{k-1} + Xe)$ 4. Maliyet fonksiyonunu minimize edecek μ değerini seç 5. $\alpha_k = \alpha_{k-1} + \mu(e_s - \alpha_{k-1})$ 6. $r^k = x - D\alpha^k$ 7. Eğer $\ \alpha_k - \alpha_{k-1}\ _2 < t$ algoritmayı durdur. Değilse Adım 1'e dön.

4.2.6. LLC

Wang ve diğerleri tarafından VQ yaklaşımlarının SPM ile kullanımına alternatif olarak önerilmiştir [49]. Algoritmanın kullandığı en küçükleme fonksiyonu Eş. 4.11'de tanımlanmıştır.

$$f(\alpha) = \frac{1}{2} \|x - D\alpha\|_2^2 + \lambda \|p \odot \alpha\|_2 \quad (4.11)$$

Burada seyrek yaklaşım için kullanılan standart l_2 -norm'da düzenleme teriminde katsayı bir p yerel adaptör ile noktasal çarpılmaktadır. Bu değişken Eş. 4.12'de tanımlanmıştır.

$$p_i = \exp\left(\frac{\text{dist}(x_i, D)}{\sigma}\right) \quad (4.12)$$

Burada p değişkeni, giriş vektörünün benzer olduğu atomların kullanımını arttıracak şekilde katsayıyı ağırlıklandırır. Böylece enküçükleme işleminde seyreklikten çok bölgeselliğin önemi arttırılmış olur.



5. ÖNERİLEN YÖNTEMLER

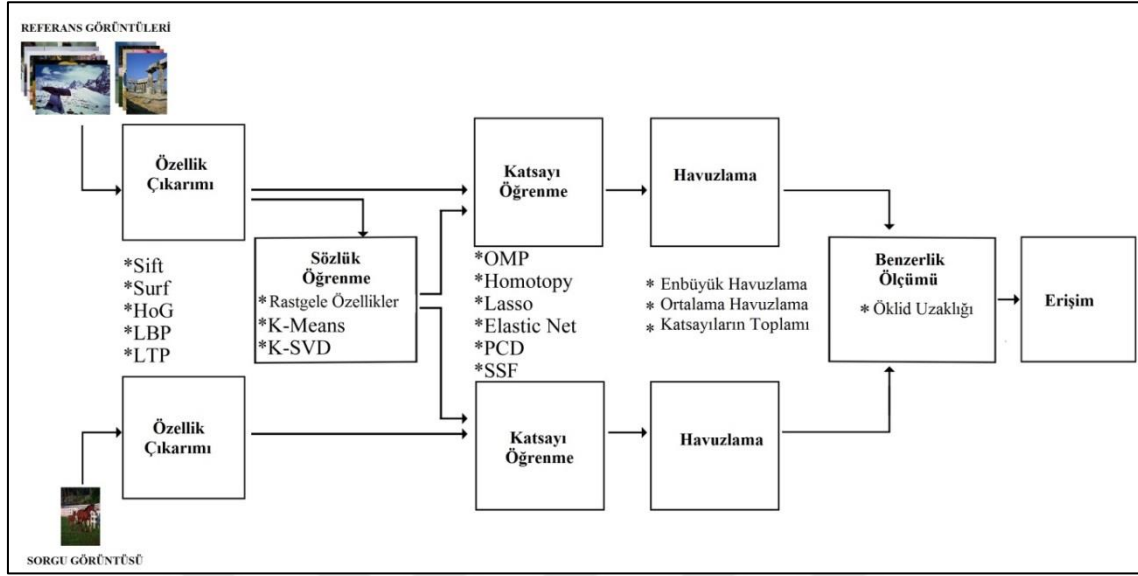
Bu çalışmada, yapılan literatür incelemesi sonucunda CBIR yaklaşımları için önerilen SR çözümlerinin, CBIR modellerindeki benzerlik ölçümlerindeki problemleri giderme amacıyla son yıllarda yoğun olarak kullanıldığı gözlemlenmiştir. Buna ek olarak LFD kullanımının literatürdeki diğer özelliklerin kullanımına göre daha etkili olduğu tespit edilmiş ve görüntü erişimlerinde özellikle SR ile kullanıldığı gözlemlenmiştir. Bu amaçla ilk olarak literatürde önerilen yaklaşımlarda kullanılan LFD ve SR algoritmalarının performansını değerlendirmek amacıyla bir CBIR çatısı önerilmiştir. Daha sonra anlamsal boşluk problemini gidermek için iki adet RBIR modeli önerilmiştir. RBIR modelleri ile CBIR modelleri arasındaki tek fark görüntüler üzerinde bölümlenme işlemi yapılmasıdır. Bu çalışmada LFD özelliklerini kümelemek için bölümlenme işlemine ihtiyaç duyulmaktadır. Bu nedenle kümeleme işlemindeki maliyeti indirmek için yeni bir yaklaşım önerilmiştir.

5.1. İçerik Tabanlı Görüntü Erişim Çatısı

İlk olarak CBIR modelleri oluşturmak için kullanılan LFD ve SR algoritmalarının performansını incelemek için bir çatı önerilmiştir. Önerilen çatı dört adımdan oluşur. Önerilen çatının adımları ve literatürdeki SR ile CBIR yaklaşımlarına göre incelenecek algoritmalar her bir adım için şu şekildedir:

- Özellik çıkarımı: SIFT, SURF, HoG, LBP ve LTP özellikleri giriş uzayı olarak kullanılmıştır.
- Sözlük öğrenme: Rastgele özellikler, K-Means ve K-SVD sözlük oluşturmak için kullanılmıştır.
- Katsayı öğrenme: OMP, Homotopy, Lasso (LARS), Elastic net, SSF ve PCD seyrek katsayı elde etmede kullanılmıştır.
- Havuzlama işlemi: İlk olarak bir resim için her bir özellikten elde edilen katsayılar enbüyük havuzlama, ortalama havuzlama ve katsayıların toplamı yöntemleri ile birleştirilir.
- Benzerlik ölçümü: Son olarak Öklid uzaklığı referans ve test görüntüleri arasındaki benzerlik ölçümü için kullanılır.

Önerilen çatı Şekil 5.1’de gösterilmektedir.



Şekil 5.1. İçerik tabanlı görüntü erişimi çatısı

5.2. Yerel Özellik Tanımlayıcıları için Hızlandırılmış Görüntü Kümeleme

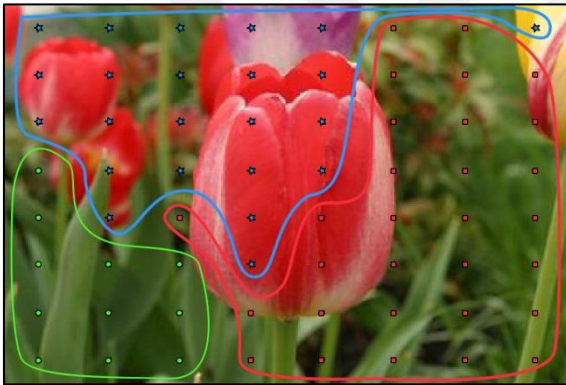
Giriş bölümünde de bahsedildiği üzere, sınıflandırma ve erişim gibi bilgisayarla görme problemlerinde görüntünün bölgesel çalışılması ve görüntü üzerindeki nesnelere kullanıldığı yapılar artmıştır. Bu çalışmalarda tüm görüntü üzerinden çıkarılan LFD noktaları, görüntüdeki bölgelere göre kümelenebilmektedir. Bu amaçla öncelikle görüntü üzerinde bölümlendirme işlemi yapıp daha sonra çıkarılan LFD vektörleri bu bölgelerde olması durumuna bakılarak kümelenebilir.

Bu çalışmada görüntüyü tüm pikselleri kullanarak bölümlenmek yerine yalnızca LFD noktalarının bulunduğu yerlerdeki pikselleri kullanılarak bölümlendirme işleminin yapıp LFD vektörlerinin kümelenebilmesi önerilmiştir. Böylelikle bu ön işleme adımının hızlandırılacağı düşünülmektedir. Yaklaşımı test etmek için literatürde popüler olarak kullanılan K-Means ve Mean Shift algoritmaları önerilen yaklaşımda kullanılmıştır. Algoritmaların piksellerin yalnız renk bilgisini kullandığı yaklaşımlarının yanı sıra, renk ve konum bilgilerini kullandığı sürümleri de incelenecektir. Bölümlendirme işlemi yapılırken tüm görüntü yerine yalnızca çıkarılmış olan LFD etrafındaki belirli boyuttaki pencereler kullanılacaktır. Yapılan testlerde farklı pencere boyutları kullanılacaktır. Örnek bir resim üzerinde standart

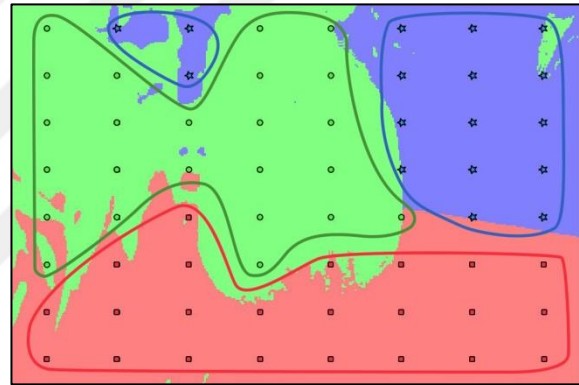
bölümleme ve önerilen yöntem ile LFD noktalarının kümelmesi Şekil 5.2’de gösterilmektedir.



a. Örnek resim



b. Önerilen yöntem ile LFD noktalarının kümelmesi



c. Standart bölümleme işlemi sonucu LFD noktalarının kümelmesi

Şekil 5.2. LFD kümeleme yaklaşımları

Önerilen yaklaşım tüm pikseller kullanılarak yapılan standart bölümleme sonuçları ile kıyaslanacaktır. Kıyaslama işlemi yaklaşımların LFD kümeleme süreleri ve kümeleme ilintilerine bakılacaktır. Kümeleme benzerliği hesaplanırken, algoritmaların işlem sonucunda LFD noktalarını rastgele etiketlemesi gibi bir problem vardır. Bir başka deyişle K-Means algoritması ile bir resim bölünüp, LFD noktaları kümelendiğinde, bu noktalara verilen etiket değerleri ile Hızlı K-Means algoritması ile kümelenen LFD noktalarının etiket değerleri farklı olabilmektedir. Bu nedenle benzerlik hesaplanmadan önce karşılaştırma yapılan algoritmalar üzerinden LFD noktaları yeniden etiketlenmiştir. Buradaki varsayım önerilen yaklaşımın da standart bölümleme algoritmaları ile benzer bir

bölümleme işlemi yapacaktır. Yöntemlerin görüntü erişimi başarımı RBIR modellerinin önerildiği bölümlerde incelenecektir.

5.3. Çevrimdışı Bölge Tabanlı Görüntü Erişim Modeli

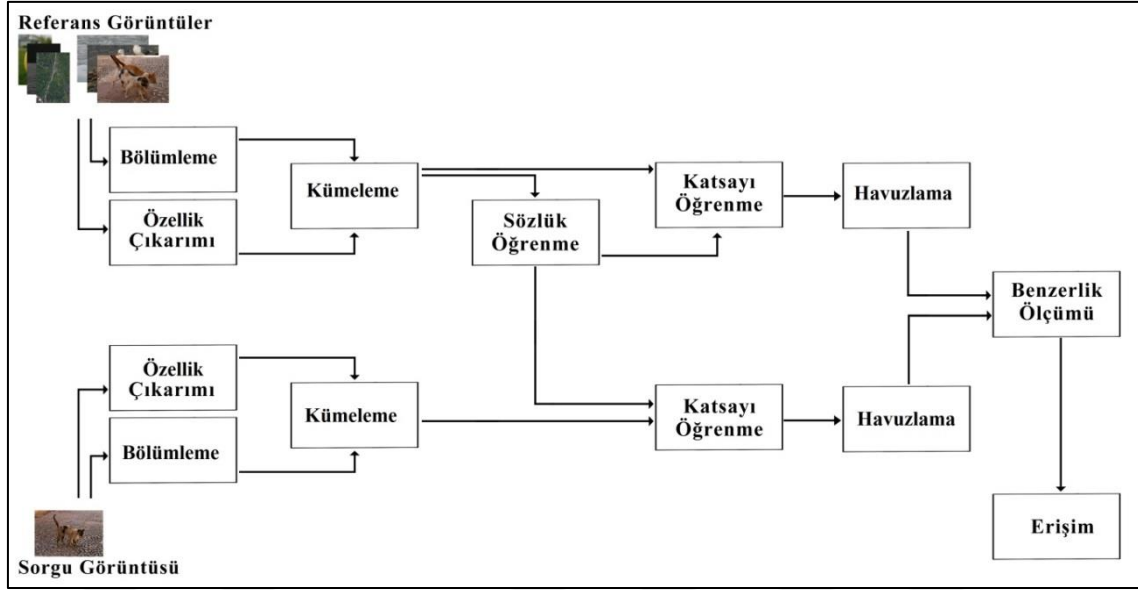
Önerilen ilk RBIR modeli çevrimdışı çalışır. Bir başka deyişle sözlük öğrenme adımında K-Means algoritması ile sözlük oluşturulmuştur. Daha sonra tekrarlayan daralma tabanlı katsayı öğrenme algoritması seyrek katsayıları sözlük üzerinde üretmek için kullanılmıştır. Önerilen yaklaşımın özgün yanı özellik çıkarımı ve seçimi adımıdır. Buna ek olarak ilk kez SR ile yapılan bir görüntü erişimi modelinde bölümleme işlemi de yapılarak SR tabanlı RBIR modeli oluşturulmuştur. Özellik çıkarımı ve seçimi adımında yerel özellikler tüm görüntü üzerinden çıkarıldıktan sonra, görüntü üzerinde yapılan bölgelere göre kümelenmektedir. Daha sonra kümelenen bu özelliklerin bölgesel ortalama ve varyans değerleri kullanılmaktadır. Böylece hem veri boyutu büyük ölçüde azaltılarak modelin daha hızlı öğrenmesi sağlanmış hem de görüntü yerine görüntüdeki bölgeler sözlük üzerinde temsil edilerek benzer bilgilerin bir arada olması sağlanmıştır.

Önerilen yaklaşımda her bir aşamada yapılan işlemler ayrıntılı olarak aşağıdaki gibidir:

- **Özellik Çıkarımı:** İlk olarak görüntülerden LFD noktaları çıkarılır. Önerilen modelde HoG, LBP_{8,1} ve LTP_{8,1} özellikleri kullanılmaktadır. Görüntüden çıkarılan her bir HoG özelliği 144 boyutlu bir vektörden oluşmaktadır. LBP_{8,1} özellikleri için 8x8 bloklardan çıkarılan özellikler 59 boyutlu histogram ile ifade edilir. Böylelikle bir görüntüden 64x59 boyutlu özellikler çıkarılmış olur. LTP_{8,1} için ise aynı blok sayısı için iki histogram elde edildiğinden 128x59 boyutlu özellikler çıkarılmış olur. Sonraki adımlar için modelde kullanılan parametrelerin boyutları LBP_{8,1} özelliklerine göre örneklendirilecektir.
- **Bölümleme:** Özellik çıkarımından sonra görüntülerde bölümleme işlemi yapılmaktadır. Önerilen modelde bir önceki yaklaşımda incelenen kümeleme-tabanlı K-Means ve Mean Shift algoritması kullanılmaktadır. Bu işlem hem referans hem de sorgu görüntüleri için gerçekleştirilmektedir. Bölümleme işlemi yapılırken görüntü piksellerinin yalnızca renk bilgilerinin yanı sıra hem renk hem de konum bilgilerinin kullanıldığı sürümü de çalışmada kullanılmıştır.

- Kümeleme: Görüntülerden özellik çıkarımı ve bölümlenme işlemi yapıldıktan sonra çıkarılan özellikler eşleştiği bölgelere göre kümelenir. Kümeleme işlemi yapıldıktan sonra her bir bölgedeki özelliklerin ortalama ve varyans değerleri hesaplanmaktadır. LBP özellikleri için 59 boyutlu ortalama ve varyans değerleri birleştirilerek 118 boyutlu bir vektör elde edilmektedir. Böylece bir görüntü $n \times 118$ boyutlu bir matrisle temsil edilmiş olur. Buradaki n sayısı görüntüdeki bölge sayısıdır ve her görüntüde farklı sayıda bölge olabilir.
- Sözlük Öğrenme: Kümeleme işlemi yapıldıktan sonra referans görüntüler kullanılarak sözlük öğrenme işlemi gerçekleştirilir. Burada artık görüntüler yerine bölgeler kullanılmaktadır. LBP özellikleri için her bir bölge, o bölgede bulunan özelliklerin ortalama ve varyans değerlerinden oluşan 118 boyutlu vektörlerden oluşmaktadır. Sözlük öğrenmesinde K-Means algoritması kullanılmaktadır. Sözlük boyutu 512 olarak alınmıştır. Öğrenme adımı sonunda 512×118 boyutlu sözlükler elde edilmektedir.
- Katsayı Öğrenme: Referans görüntüleri üzerinden sözlük öğrenimi gerçekleştirildikten sonra görüntüdeki bölgeleri temsil eden vektörlerin bu sözlüğe göre seyrek katsayıları hesaplanmaktadır. Bu çalışmada tekrarlayan-daralma tabanlı SSF katsayı öğrenme algoritması kullanılmaktadır. Bu algoritma daha önce literatürde RBIR çalışmalarında kullanılmamıştır. Öğrenme sonucunda n adet bölgeye sahip bir görüntü için $n \times 512$ boyutlu seyrek vektörlerden oluşan bir matris elde edilmektedir.
- Havuzlama: Elde edilen seyrek katsayılar ortalama havuzlama, enbüyük havuzlama ve katsayıların toplamı kullanılarak tek bir vektöre dönüştürülmektedir. Burada bir görüntü için elde edilen $n \times 512$ boyutlu seyrek katsayıların satırlarının ortalaması alınarak 512 boyutlu tüm görüntüyü temsil eden tek bir vektör elde edilmektedir.
- Benzerlik Ölçümü: Havuzlama işlemi yapıldıktan sonra her bir görüntü bir vektörle temsil edilmektedir. Benzerlik ölçümü adımında sorgu görüntüsü ile tüm referans görüntüleri arasında Öklid uzaklığı hesaplanmaktadır. Bu uzaklığa göre referans görüntüleri küçükten büyüğe sıralanmaktadır.
- Erişim: Sorgu görüntüsü ile benzerliği hesaplanmış ve sıralanmış referans görüntülerinden ilk N sayıdaki görüntü listelenmektedir.

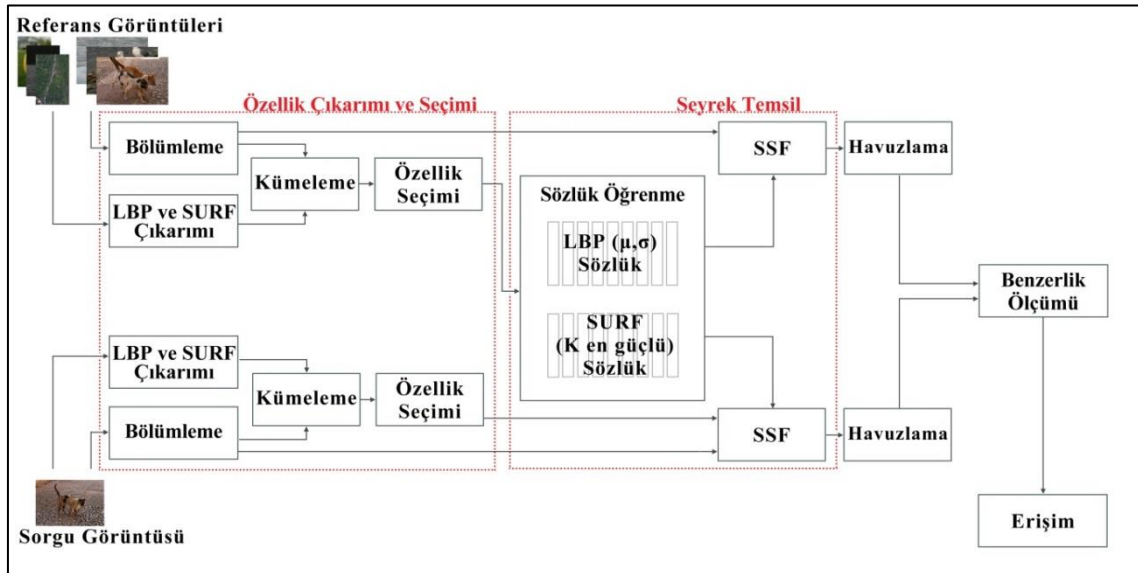
Önerilen yaklaşımın akış şeması Şekil 5.3'te gösterilmektedir.



Şekil 5.3. Çevrimdışı RBIR modeli

5.4. Çevrimiçi Bölge Tabanlı Görüntü Erişim Modeli

Bir önceki bölümde önerilen çevrimdışı RBIR modelini geliştirmek adına sözlük öğrenme algoritması ve farklı özelliklerin kullanımına odaklanılmıştır. Önerilen ikinci model Şekil 5.4'te gösterilmektedir.



Şekil 5.4. Çevrimiçi RBIR modeli

SURF noktaları birbirlerine göre güçlünden zayıfa doğru sıralanabilen noktalardır. Burada LBP noktalarının yanı sıra belirli sayıdaki SURF noktası bir bölgeyi sözlük üzerinde temsil edebilir. Bununla birlikte K-SVD algoritması, sözlük öğrenimi ve katsayı öğrenimini ardışık olarak yürüterek sözlük oluşturmada oldukça etkili bir algoritmadır. Burada sözlük, oluşturulacak referans eğitim kümesinin etiketleri bilindiği için sözlük üzerinde her bir etiket belirli bir konuma konumlandırılabilceği düşünülmektedir. Algoritmanın adımları aşağıdaki gibidir.

- **Özellik Çıkarımı:** İlk olarak görüntülerden LBP noktaları çıkarılır. Bu çalışmada $LBP_{8,1}$ ve SURF özellikleri kullanılmaktadır. İlk olarak görüntü 4×4 ve 8×8 pencereye bölünmüş ve her bir pencereden 59 boyutlu LBP özellikleri çıkarılmıştır. Böylece her bir görüntü için 16×59 ve 64×59 boyutlu özellikler elde edilmiştir. Bu işlem hem referans hem de sorgu görüntüleri için yapılmıştır. Buna ek olarak tüm görüntüden SURF noktaları çıkarılmıştır.
- **Bölümleme:** Sonraki adımda ise görüntülerde bölümleme işlemi yapılmaktadır. Önerilen modelde yine çevrimdışı modelde olduğu gibi kümeleme-tabanlı K-Means ve Mean Shift algoritmaları kullanılmıştır. Bu işlem hem referans hem de sorgu görüntüleri için gerçekleştirilmiştir. Bölümleme işlemi yapılırken görüntü piksellerinin yalnız renk bilgilerinin kullanımının yanı sıra hem renk hem de konum bilgileri de birlikte kullanılmıştır. Böylelikle her bir görüntü farklı sayıda bölgelere ayrılmıştır.
- **Kümeleme:** Görüntülerden özellik çıkarımı ve bölümleme işlemi yapıldıktan sonra çıkarılan özellikler eşleştiği bölgelere göre kümelenebilmektedir. Kümeleme işlemi yapıldıktan sonra LBP için her bir bölgedeki özelliklerin ortalama ve varyans değerleri hesaplanır. 59 boyutlu bu değerler birleştirilerek 118 boyutlu bir vektör elde edilir. Böylece bir görüntü $n \times 118$ boyutlu matrislerle temsil edilmiş olur. Buradaki n sayısı görüntüdeki bölge sayısıdır ve her görüntüde farklı sayıda bölge olabilir. Daha sonra SURF noktaları için en güçlü K nokta her bir bölge için belirlenir ve $n \times K \times 128$ boyutlu matris elde edilmiş olur. Böylelikle bir görüntü $n \times 118$ boyutlu LBP ve $n \times K \times 128$ boyutlu SURF matrisleri ile temsil edilmiş olur. Bu işlem test görüntüleri için de gerçekleştirilecektir.
- **Sözlük Öğrenme:** Kümeleme işlemi yapıldıktan sonra referans görüntüler kullanılarak sözlük öğrenme işlemi gerçekleştirilir. Burada artık görüntüler yerine bölgeler kullanılmaktadır. Her bir bölge, o bölgede bulunan LBP noktalarının ortalama ve

varyans değerlerinden oluşan 118 boyutlu vektörlerden ve $K \times 128$ boyutlu SURF noktalarından oluşmaktadır. Sözlük boyutu LBP için kullanılan veri tabanındaki etiket sayısı (ES) kadar tanımlanmıştır. SURF için ise $K \cdot ES$ olarak alınmıştır. Böylece öğrenme adımı sonunda $ES \times 118$ ve $(K \cdot ES) \times 128$ boyutlu iki sözlük elde edilecektir. Burada K-SVD sözlüğü eğitimi gerçekleştirilirken, referans eğitim veri tabanındaki görüntülerin etiketleri de beklenen katsayı olarak eğitime dâhil edilmiş ve böylece sözlüğün belirli bir bölgede eldeki veri tabanı görüntülerini kümelemesi amaçlanmıştır. K-SVD algoritması sözlüğü oluşturmak için Eş. 5.1'deki maliyet fonksiyonunu enküçükler.

$$\langle D, \alpha \rangle = \arg \min_{D, \alpha} \|x - D\alpha\|_2^2 \text{ s.t. } \forall i, \|\alpha_i\|_0 \leq k \quad (5.1)$$

Burada x özellik matrisi, D sözlük, α seyrek katsayılar ve k sıfır olmayan atom sayısıdır. Bu çalışmada Eş. 5.1'de tanımlanan maliyet fonksiyonuna, etiket değişkeni L eklenmiştir. Bu değişken veri kümesinin etiketlerine göre belirlenir. Böylelikle, benzer özelliklerin seyrek katsayıları sözlük üzerinde benzer bölgelerde konumlandırılacaktır. Bu değişkenle birlikte yeni maliyet fonksiyonu Eş. 5.2'de tanımlandığı gibidir.

$$\langle D, \alpha \rangle = \arg \min_{D, \alpha} \|x - D(L \odot \alpha)\|_2^2 \text{ s.t. } \forall i, \|(L \odot \alpha_i)\|_0 \leq k \quad (5.2)$$

Burada \odot skalar çarpımdır. Etiket değişkeni sayesinde benzer etiketli görüntüler sözlük üzerinde benzer bölgelere konumlandırılırken, farklı görüntüler sözlük üzerinde birbirinde uzak bölgelerde konumlandırılır. Önerilen yaklaşım çok etiketli veri kümeleri üzerinde çalıştırılacağından L değişkenini belirlemek oldukça kritiktir. m adet n boyutlu vektörlerden oluşan bir X özellik kümesi l farklı etiket alabilsin. Bu durumda özellik kümesi X ($n \times m$), sözlük D ($n \times l$) ve etiket değişkeni L ($l \times m$) boyutlu olacaktır. Çok etiketli veri kümesi için örnek bir etiket matrisi Eş. 5.3'dedir.

$$O \equiv \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (5.3)$$

Eğer sözlük boyutu veri kümesinin etiket boyutuna eşit ise, eşitlik değişkeni L , Eş. 5.3'te tanımlanan etiket kümesine göre Eş. 5.4'teki değerleri alır.

$$L \equiv \begin{bmatrix} 0 & 0,33 & 0 & 0,5 \\ 0,5 & 0,33 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0,5 & 0,33 & 0 & 0,5 \end{bmatrix} \quad (5.4)$$

Burada sözlük boyutunun, veri kümesi etiket boyutuna eşit olduğu varsayılmıştır. Böylece, sözlükteki her bir sütun, bir etiketi temsil etmiş olur. Sözlük boyutu veri kümesi etiket boyutuna göre orantısal olarak artarsa, bu durumda her bir bölge bir etiketi temsil edecektir. Bu adımda iki sözlük önerilen maliyet fonksiyonu ile eğitilecektir. Sözlüklerden biri LBP ortalama ve varyans vektörlerinden oluşurken, diğeri SURF özelliklerinden oluşmaktadır. Her bir bölge, bir LBP ortalama ve varyans vektörü ile temsil edildiğinden, LBP için oluşturulan sözlük boyutu veri kümesi etiket boyutuna eşittir. Diğ taraftan her bir bölge, K adet SURF özelliği ile temsil edildiğinden, SURF özellikleri için oluşturulan sözlük boyutu ($K \cdot l$)'dir.

- **Katsayı Öğrenme:** Referans görüntüleri üzerinden sözlük öğrenimi gerçekleştirildikten sonra görüntüdeki bölgeleri temsil eden vektörlerin bu sözlüğe göre seyrek katsayıları hesaplanmıştır. Önerilen çevrimiçi RBIR modelinde de yine tekrarlayan-daralma tabanlı SSF katsayı öğrenme algoritması kullanılmıştır. Öğrenme sonucunda LBP için n adet bölgeye sahip bir görüntü için $n \times ES$ boyutlu seyrek vektörlerden oluşan bir matris elde edilir. SURF için ise $n \times (K \cdot ES)$ boyutlu seyrek matris elde edilir.
- **Havuzlama:** Elde edilen seyrek katsayılar ortalama havuzlama, enbüyük havuzlama ve katsayıların toplamı kullanılarak tek bir vektöre dönüştürülür. Burada bir görüntü için elde edilen LBP özellikleri üzerinden $n \times ES$ boyutlu seyrek katsayıların satırlarının ortalaması alınarak ES boyutlu tüm görüntüyü temsil eden tek bir vektör elde edilir. SURF noktaları için de yine benzer şekilde $n \times (K \cdot ES)$ seyrek katsayılar her bir bölge için birleştirilerek $n \times (ES)$ boyutlu olarak oluşturulur ve daha sonra tüm görüntü için ES boyutlu tüm görüntüyü temsil eden tek bir vektör havuzlama yöntemi ile elde edilir.
- **Benzerlik Ölçümü:** Havuzlama işlemi yapıldıktan sonra her bir görüntü bir SURF ve LBP vektörüyle temsil edilmektedir. Benzerlik ölçümü adımında sorgu görüntüsü ile tüm referans görüntüleri arasında Öklid uzaklığı hesaplanır. Bu uzaklığa göre referans

görüntüleri küçükten büyüğe sıralanır. Benzerlik LBP ve SURF benzerlikleri olarak hesaplandıktan sonra nihai benzerlik aşağıdaki gibi hesaplanır.

$$Sim = t_1 * Sim_{lbp} + (1 - t_1) * Sim_{surf} \quad (5.5)$$

- Erişim: Sorgu görüntüsü ile benzerliği hesaplanmış ve sıralanmış referans görüntülerinden ilk N sayıdaki görüntü listelenir.



6. DENEYSEL SONUÇLAR

Bu bölümde, önerilen yaklaşımlar test edilmiş ve literatürdeki SR tabanlı CBIR yaklaşımları ile kıyaslanmıştır. İlk olarak Bölüm 6.1’de, kullanılan veri kümeleri ve başarımlar değerlendirme ölçütleri açıklanmıştır. Bölüm 6.2’de, önerilen CBIR çatısı sonuçları ve elde edilen sonuçların literatürdeki SR tabanlı CBIR yaklaşımları ile karşılaştırmaları yapılmıştır. Bölüm 6.3’te, önerilen LFD kümeleme yaklaşımının standart yaklaşımla karşılaştırmalı sonuçları verilmiştir. Bölüm 6.4 ve Bölüm 6.5’te ise sırasıyla çevrimdışı ve çevrimiçi RBIR modelleri test edilmiş ve literatürdeki SR tabanlı CBIR modelleri ile karşılaştırmalı sonuçları verilmiştir.

6.1. Veri Kümeleri ve Performans Değerlendirme Ölçütleri

İlk olarak önerilen yöntemlerin testleri için kullanılan veri kümeleri ve performans değerlendirme ölçütleri bu bölümde anlatılacaktır.

6.1.1. Veri kümeleri

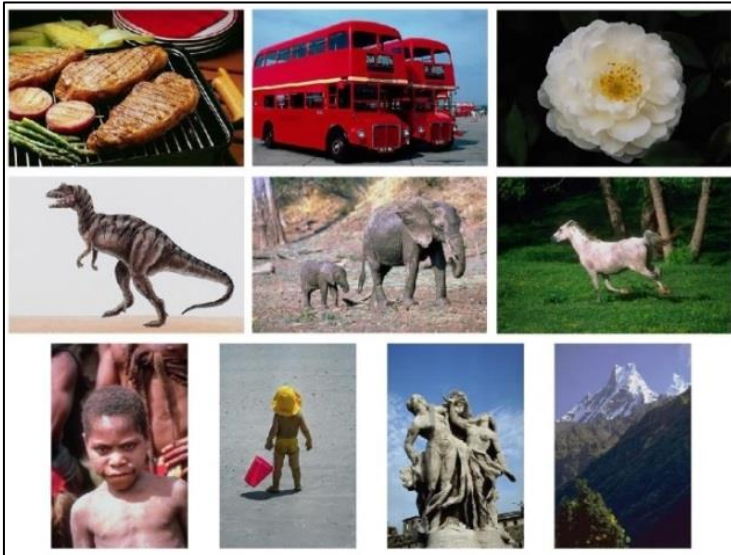
Bu çalışmada dört veri kümesi yöntemlerin performanslarını değerlendirmede kullanılacaktır. Bu veri kümeleri Coil 20 [26], Corel 1000 [27], Photography [28] ve ImageCLEF 2008 [29] veri kümeleridir. Coil 20 ve Corel 1000 veri kümeleri tek etiketlerden oluşmaktadır. Coil 20 veri kümesi 1440 görüntüden oluşmaktadır. Bu görüntüler 20 objenin 72 farklı açıdan elde edilmiş koleksiyonundan oluşmaktadır. Her bir görüntü 128x128 boyutlu ve gri seviyedir. Her bir objeden 6 adet olacak şekilde 120 adet görüntü sorgu için, kalanlar ise veri tabanı olarak kullanılacaktır. Veri tabanındaki objeler Şekil 6.1’dedir.

Corel 1000 veri kümesi 10 sınıftan oluşan 1000 adet görüntü içerir. Veri kümesinin sınıfları Şekil 6.2’de gösterilmektedir. Her sınıftan 10 görüntü olacak şekilde 100 görüntü sorgu, kalanlar ise veri tabanı olarak kullanılmaktadır. Görüntüler farklı boyutlarda ve renkli seviyedir.

Photography ve ImageCLEF 2008 veri kümeleri çoklu etiketlerden oluşmaktadır ve görüntü erişimi ile otomatik ek açıklama işlemlerinde yaygın olarak kullanılmaktadır. Önerilen RBIR modellerinin ve LFD kümeleme algoritmalarının başarımlerinde kullanılacaktır. Önerilen çalışmalar bölgesel tabanlı olduğu için çoklu etiketli veri kümeleri ile yapılacak karşılaştırma işleminin daha hassas sonuçlar vereceği düşünülmüştür.



Şekil 6.1. Coil 20 veri tabanı objeleri



Şekil 6.2. Corel 1000 veri tabanı sınıfları

Photography veri kümesi 2360 görüntüden oluşmaktadır. Bu görüntüler farklı boyutlarda ve renkli seviyededir. Görüntülerin üzerinde 141 farklı etiket bulunmaktadır. Çalışmada

2118 görüntü eğitim, 242 görüntü ise test için kullanılmıştır. Veri kümesindeki örnek resimler Şekil 6.3’de gösterilmektedir. ImageCLEF 2008 veri kümesi, etiketi bildirilen 1827 eğitim verisi ve etiketi bildirilmemiş 1000 test verisinden oluşmaktadır. Bu çalışma da etiketli resimlerden 1645 adet görüntü eğitim, 182 adet görüntü ise test için ayrılmıştır. 1000 adet test görüntüsü etiketleri olmadığı için kullanılmamıştır. Veri kümesi farklı boyutlarda ve renkli seviyededir. Görüntü kümesi hiyerarşik 17 farklı etiketten oluşmaktadır. Veri kümesindeki örnek resimler Şekil 6.4’te gösterilmektedir.



Şekil 6.3. Photography veri tabanı



Şekil 6.4. ImageCLEF 2008 veri tabanı

6.1.2. Performans değerlendirme ölçütleri

Kesinlik ve hatırlama görüntü erişimi sistemlerinin performans değerlendirmelerinde geniş ölçüde kullanılmaktadır. Bu ölçümler aşağıdaki gibi hesaplanır.

$$\text{kesinlik} = \frac{\text{Erişilen ilgili görüntülerin sayısı}}{\text{Tüm erişilen görüntü sayısı}} \quad (6.1)$$

$$\text{hatırlama} = \frac{\text{Erişilen ilgili görüntülerin sayısı}}{\text{Tüm ilgili görüntü sayısı}} \quad (6.2)$$

MAP ilk N sıralama için tüm sorguların kesinlik değerlerinin ortalaması anlamına gelir ve şu şekilde hesaplanır.

$$MAP = \frac{1}{Q} \sum_{i=1}^Q \left(\frac{\sum_{r=1}^N P(r)}{\text{Tüm ilgili görüntü sayısı}} \right) \quad (6.3)$$

Burada Q sorgu sayısı, r rank ve N i. sorgu için ilgili görüntü sayısıdır. Cumulative Match Characteristic (CMC) (Birikimli Eşleştirme Karakteristiği - BEK) destekleyici bir değerlendirme ölçümüdür [89-91]. Özellikle 1:n tanımlama sistemlerinin değerlendirilmesinde kullanışlıdır. Bu ölçüm, en iyi doğru eşleşme yerine, en iyi n sıralama içindeki doğru cevaplarla ilgilenir. Bu çalışmada önerilen CBIR çatısı testleri için MAP, Kesinlik-Hatırlama eğrisi ve CMC eğrisi performans değerlendirmesinde kullanılacaktır. Çoklu etiketli görüntülerin karşılaştırmaları için yapılan işlemlerde standart kesinlik ve hatırlama hesaplamaları ve dolayısıyla MAP değerleri doğru başarımlar ölçütleri değildir [92, 93]. Bu nedenle literatürde yaygın olarak Normalized Discounted Cumulative Gain (NCDG) (Normalleştirilmiş Azaltılmış Birikimli Kazanç - NABK) yöntemi kullanılmaktadır [94, 95]. Buna ek olarak literatürde dereceli kesinlik, dereceli hatırlama ve Graded MAP (GMAP) (Dereceli OKO – DOKO) hesaplamaları çoklu etiketli veri kümeleri kullanan modellerin karşılaştırılmasında başarımlar ölçütü olarak önerilmiştir [96]. Bu çalışmada NCDG ölçütünün yanı sıra bu ölçütler de çoklu etiketli veri kümelerinin kullanıldığı testlerde kullanılacaktır.

6.2. CBIR Çatısı Karşılaştırmalı Sonuçları

Çalışmada ilk olarak SR algoritmaları önerilen CBIR çatısı ile çalıştırılmış ve farklı parametrelerle elde edilen sonuçları sunulmuştur. Daha sonra algoritmaların CBIR çatısı ile elde ettiği en iyi birleşimler literatürdeki SR tabanlı çalışmalarla kıyaslanmıştır [32, 49, 57]. Bu algoritmalar Çalışma 30, Çalışma 32, Çalışma 37 ve Çalışma 43'te kullanılmıştır.

6.2.1. Parametreler

Önerilen çatının her bir adımında kullanılan algoritmalar ve bu algoritmaların parametreleri aşağıdaki gibidir:

- LFD algoritmaları: SIFT özelliklerinin çıkarılması için Lowe tarafından hazırlanan kaynak kod kullanılmıştır. Tüm parametrelere koddaki varsayılan değerler atanmıştır. Standart Matlab fonksiyonları SURF ve HoG özelliklerini elde etmek için kullanılmıştır. SURF noktalarını elde etmek için SURFSize parametresi 128 olarak belirlenmiş ve diğer parametreler ise varsayılan değer olarak atanmıştır. HoG özelliklerini elde etmek için Block-Size=[4 4] olarak belirlenmiş ve diğer parametreler fonksiyonun varsayılan değerleri olarak belirlenmiştir. LBP8,1 ve LTP8,1 algoritmaları LBP ve LTP özelliklerini çıkarmak için kullanılmıştır. Burada her bir pikseli etiketlemek için etrafındaki 8 komşusunun bilgisi kullanılmıştır.
- Sözlük öğrenme algoritmaları: Coil 20 veri kümesi üzerinde tüm algoritmalar için sözlük boyutu 256 ve 512 olarak belirlenmiştir. Elde edilen sonuçlar sözlük boyutunun erişim kesinliği üzerinde belirgin bir fark olmadığını gösterdiğinden Corel 1000 veri kümesi üzerinde sözlük boyutu yalnız 512 olarak belirlenmiştir. Tüm algoritmalar 100 yineleme için eğitilmiştir.
- Katsayı öğrenme algoritmaları: OMP algoritması için sıfır olmayan atom sayısı 1 olarak belirlenmiştir. Homotopy algoritması 100 yineleme için eğitilmiş ve λ parametresi $1e-6$ olarak belirlenmiştir. Lasso ve Elastic net algoritmaları için standart Matlab fonksiyonları kullanılmıştır. Lasso algoritması için Alpha parametresi 1, DFmax parametresi ise 3 olarak belirlenmiş, diğer parametrelerin varsayılan değerleri kullanılmıştır. Elastic net algoritması için Alpha parametresi 0.5, DFmax parametresi 3 olarak belirlenmiş, diğer parametrelerin varsayılan değerleri kullanılmıştır. SSF ve PCD algoritmaları 10 yineleme için eğitilmiş ve λ parametresi 0.01 olarak belirlenmiştir. LLC

algoritması için λ parametresi $1e-4$, knn parametresi 5 ve pyramid parametresi [1, 2, 4] olarak belirlenmiştir. Algoritma 100 yineleme için eğitilmiştir. Çalışma [32]'de kullanılan algoritma ise 50 yineleme için eğitilmiştir.

6.2.2. Coil 20 veri kümesi sonuçları

Çizelge 6.1'de Rastgele Özellikler ve K-Means sözlük algoritmasının farklı havuzlama yöntemleri ile elde edilen test sonuçları gösterilmektedir.

Çizelge 6.1. Coil 20 üzerinde 512 boyutlu sözlükler için MAP değerleri

		Rastgele Özellikler					K-Means				
		SIFT	SURF	HoG	LBP	LTP	SIFT	SURF	HoG	LBP	LTP
Ortalama Havuzlama	OMP	0,45	0,33	0,31	0,71	0,76	0,56	0,44	0,31	0,72	0,76
	Homotopy	0,65	0,47	0,67	0,75	0,76	0,69	0,53	0,71	0,74	0,83
	Lasso	0,54	0,45	0,63	0,77	0,78	0,48	0,46	0,63	0,78	0,81
	Elastic net	0,54	0,45	0,63	0,81	0,78	0,48	0,46	0,63	0,8	0,84
	SSF	0,66	0,52	0,67	0,79	0,78	0,67	0,51	0,65	0,79	0,8
	PCD	0,66	0,52	0,67	0,79	0,78	0,68	0,51	0,65	0,79	0,8
Enbüyük Havuzlama	OMP	0,37	0,19	0,13	0,6	0,68	0,43	0,25	0,13	0,65	0,73
	Homotopy	0,69	0,44	0,67	0,69	0,75	0,68	0,45	0,74	0,73	0,83
	Lasso	0,41	0,29	0,55	0,7	0,78	0,37	0,31	0,61	0,71	0,82
	Elastic net	0,41	0,28	0,56	0,69	0,76	0,37	0,31	0,61	0,62	0,8
	SSF	0,69	0,57	0,68	0,77	0,74	0,73	0,57	0,7	0,76	0,75
	PCD	0,69	0,57	0,7	0,77	0,74	0,73	0,58	0,7	0,76	0,75
Katsayıların Toplamı	OMP	0,43	0,23	0,31	0,71	0,76	0,49	0,3	0,31	0,72	0,76
	Homotopy	0,68	0,45	0,67	0,75	0,76	0,65	0,46	0,71	0,74	0,83
	Lasso	0,51	0,39	0,63	0,77	0,78	0,4	0,37	0,63	0,78	0,81
	Elastic net	0,52	0,39	0,63	0,81	0,78	0,4	0,37	0,63	0,8	0,84
	SSF	0,7	0,55	0,67	0,79	0,78	0,69	0,53	0,65	0,79	0,8
	PCD	0,7	0,55	0,67	0,79	0,78	0,7	0,53	0,65	0,79	0,8

Sonuçlar incelendiğinde ilk olarak ortalama havuzlama ve katsayıların toplamının en iyi sonucu verdiği görülmektedir. Rastgele özellikler için en iyi birleşim LBP, Elastic net ve ortalama havuzlama ya da katsayıların toplamı olarak %81 MAP elde etmiştir. Diğer taraftan K-Means için en iyi birleşim LTP, Elastic net, ortalama havuzlama ya da katsayıların toplamı olarak %84 MAP elde etmiştir. İki sözlük arasındaki sonuçlar

incelendiğinde aynı katsayı öğrenme algoritması ve havuzlama yöntemi için K-Means algoritmasının daha iyi erişim başarımı yakaladığı görülmektedir.

Çizelge 6.2. Coil 20 üzerinde K-SVD sözlük için MAP değerleri

		256 Boyutlu K-SVD					512 Boyutlu K-SVD				
		SIFT	SURF	HoG	LBP	LTP	SIFT	SURF	HoG	LBP	LTP
Ortalama Havuzlama	OMP	0,06	0,45	0,59	0,68	0,76	0,57	0,42	0,58	0,68	0,74
	Homotopy	0,7	0,55	0,6	0,75	0,81	0,69	0,52	0,6	0,73	0,79
	Lasso	0,49	0,48	0,65	0,77	0,82	0,48	0,47	0,69	0,76	0,81
	Elastic net	0,49	0,49	0,65	0,78	0,83	0,49	0,47	0,69	0,77	0,82
	SSF	0,68	0,51	0,63	0,79	0,8	0,67	0,51	0,63	0,79	0,8
	PCD	0,68	0,51	0,63	0,79	0,8	0,67	0,51	0,64	0,79	0,8
Enbüyük Havuzlama	OMP	0,05	0,31	0,71	0,64	0,73	0,45	0,25	0,7	0,62	0,7
	Homotopy	0,75	0,52	0,73	0,74	0,82	0,69	0,44	0,72	0,7	0,79
	Lasso	0,31	0,31	0,62	0,7	0,82	0,39	0,31	0,65	0,65	0,81
	Elastic net	0,31	0,31	0,62	0,69	0,81	0,39	0,31	0,65	0,63	0,81
	SSF	0,71	0,57	0,67	0,76	0,74	0,72	0,57	0,67	0,77	0,74
	PCD	0,71	0,57	0,67	0,76	0,74	0,72	0,57	0,67	0,77	0,74
Katsayıların Toplamı	OMP	0,06	0,34	0,59	0,68	0,76	0,5	0,27	0,58	0,68	0,74
	Homotopy	0,69	0,51	0,6	0,75	0,81	0,66	0,45	0,6	0,73	0,79
	Lasso	0,41	0,4	0,65	0,77	0,82	0,41	0,36	0,69	0,76	0,81
	Elastic net	0,41	0,4	0,65	0,78	0,83	0,41	0,36	0,69	0,77	0,82
	SSF	0,7	0,53	0,63	0,79	0,8	0,69	0,53	0,63	0,79	0,8
	PCD	0,7	0,53	0,63	0,79	0,8	0,69	0,53	0,64	0,79	0,8

Çizelge 6.2’de 256 ve 512 boyutlu K-SVD sözlükleri için MAP değerleri gösterilmektedir. Çizelgedeki sonuçlar göstermektedir ki farklı sözlük boyutları üzerinde belirgin bir farklılık yoktur. En iyi birleşim LTP ve ortalama havuzlama ya da katsayıların toplamı olarak %83 MAP ile elde edilmiştir.

Çizelge 6.3’de ODL algoritması kullanılarak oluşturulan CBIR modelinin sonuçları gösterilmektedir. Model LTP ve ortalama havuzlama ya da katsayıların toplamı olarak %75 ile en iyi MAP değerini üretmiştir.

Çizelge 6.4’te tüm modellerin en iyi sonuçları gösterilmektedir. Buna göre literatürdeki yaklaşımlardan LLC algoritması %89 ile en iyi sonucu vermiştir.

Çizelge 6.3. Coil 20 üzerinde ODL algoritması için MAP değerleri

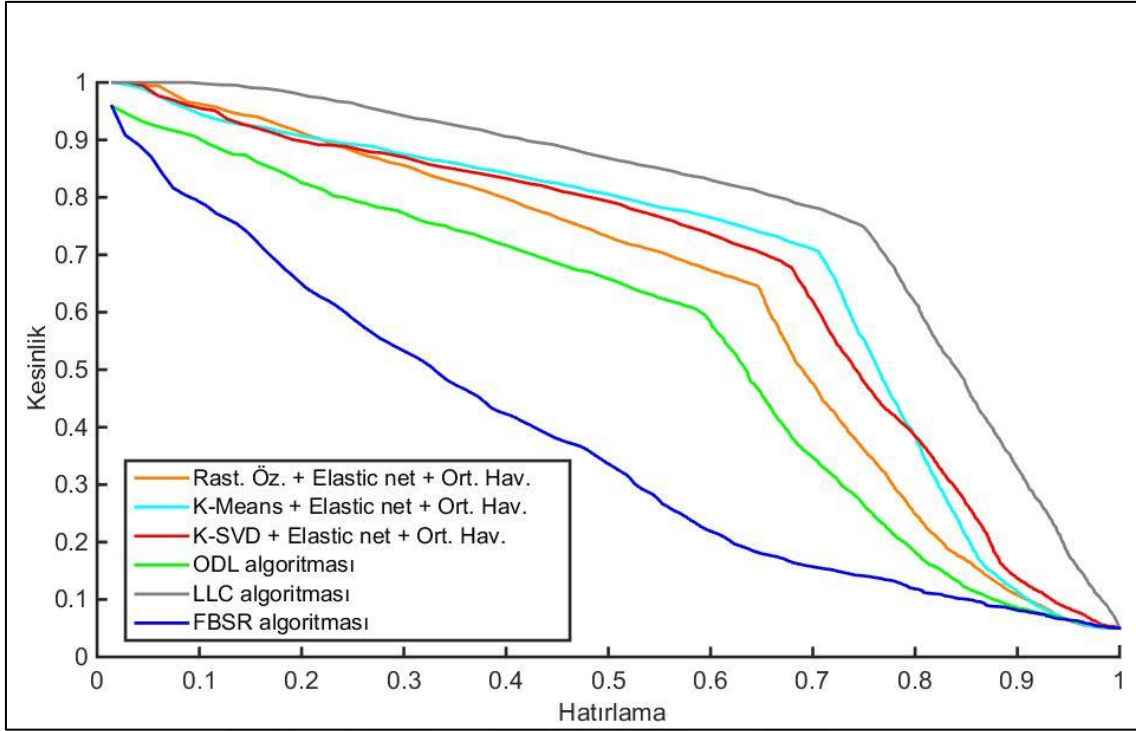
	Ortalama Havuzlama	Enbüyük Havuzlama	Katsayıların Toplamı
SIFT	0,52	0,50	0,55
SURF	0,36	0,21	0,27
HoG	0,58	0,66	0,58
LBP	0,68	0,62	0,68
LTP	0,75	0,70	0,75

Çizelge 6.4. Coil 20 için tüm yaklaşımların en iyi sonuçları

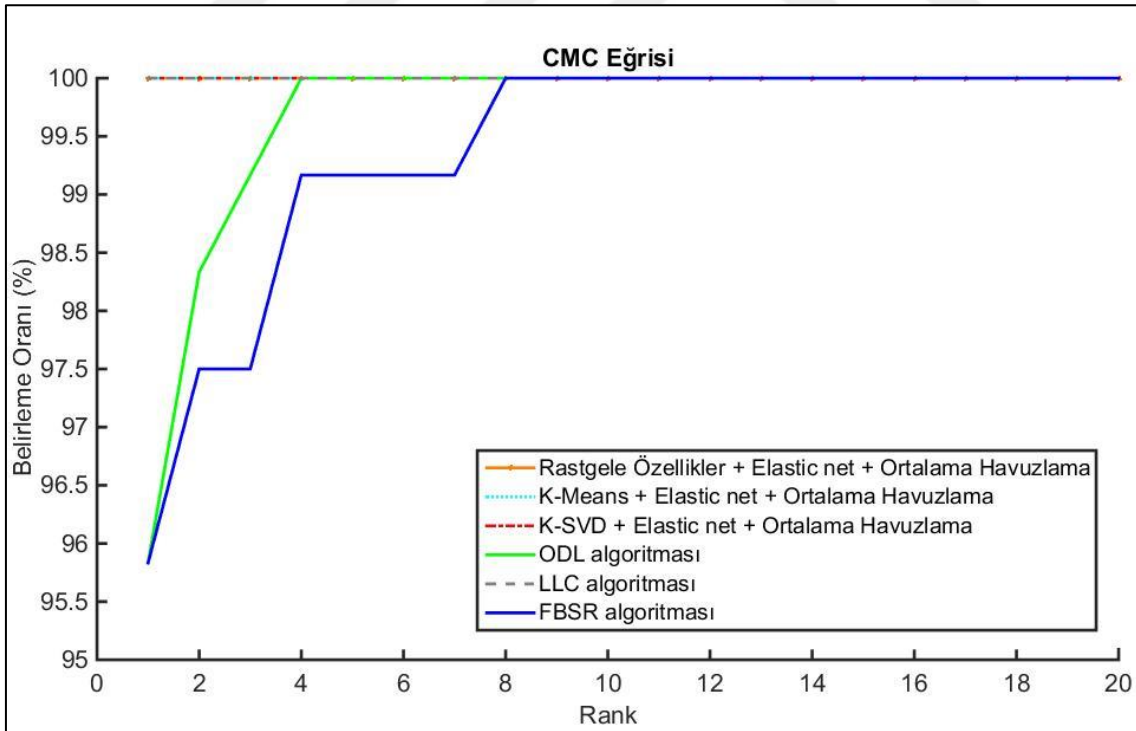
	SIFT	SURF	HoG	LBP	LTP
Rastgele Özellikler + Elastic net + Ortalama Havuzlama	0,54	0,45	0,63	0,81	0,78
K-Means + Elastic net + Ortalama Havuzlama	0,48	0,46	0,63	0,8	0,84
K-SVD + Elastic net + Ortalama Havuzlama	0,49	0,49	0,65	0,78	0,83
Çalışma [57] – ODL algoritması	0,52	0,36	0,58	0,68	0,75
Çalışma [49] – LLC algoritması	0,89	0,75	0,8	0,78	0,83
Çalışma [32] – FBSR algoritması	0,5	0,7	0,49	0,59	0,58

Şekil 6.5 ve Şekil 6.6’da Coil 20 veri kümesi için CBIR çatısı üzerindeki her bir sözlük için en iyi birleşimlerin ve literatürdeki çalışmaların kesinlik-hatırlama eğrileri ve CMC eğrileri verilmektedir. Bu eğriler incelendiğinde LLC algoritması kesinlik-hatırlama eğrisinde ideale en yakın değer olmakla birlikte CMC eğrisinde en hızlı ideale yaklaşan algoritmadır.

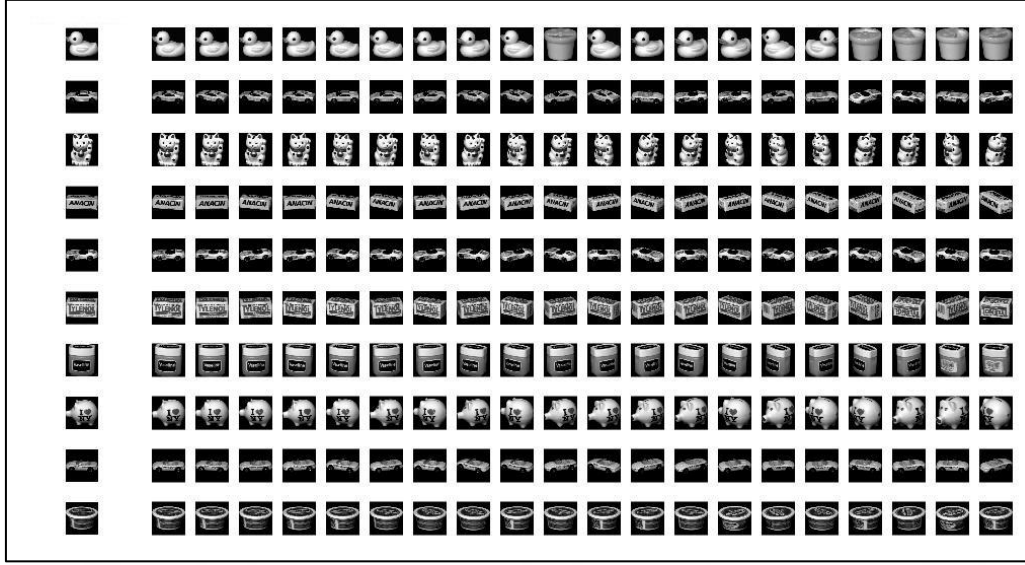
Şekil 6.7’de LLC algoritmasının Coil 20 veri kümesi üzerindeki rastgele 10 obje sorgusu için en iyi ilk 20 görüntü gösterilmektedir.



Şekil 6.5. Coil 20 için tüm yaklaşımların kesinlik-hatırlama eğrisi



Şekil 6.6. Coil 20 için tüm yaklaşımların CMC eğrisi



Şekil 6.7. Coil 20 veri kümesi üzerindeki rastgele 10 obje için en iyi ilk 20 sonuç

Çizelge 6.5. Coil 20 için tüm sözlükler üzerinde katsayı öğrenme süreleri (saniye)

		SIFT	SURF	HoG	LBP	LTP
Rasgele Özellikler	OMP	55,17	23,23	45,7	17,42	28,85
	Homotopy	4.585,53	1.756,11	1.725,82	874,41	1.622,54
	Lasso	51.140,68	20.545,46	37.474,77	10.046,43	18.928,71
	Elastic net	47.822,8	21.488,53	38.136,15	11.476,73	23.077,44
	SSF	26.846,42	10.723,99	21.887,37	7.107,32	14.005,65
	PCD	25.351,86	10.722,4	21.830,01	7.238,43	14.051,47
K-Means	OMP	46,68	22,65	42,06	16,55	26,06
	Homotopy	4.237,89	1.778,85	3.731,93	802,06	1.737,92
	Lasso	64.537,98	24.752,93	47.125,54	9.525,57	16.218,98
	Elastic net	61.312,45	24.585,78	47.228,82	10.386,88	18.047,63
	SSF	27.450,69	10.336,34	23.041,01	6.980,02	14.997,29
	PCD	25.850,34	10.911,28	20.950,28	7.014,06	13.921,62
K-SVD	OMP	48,51	22,9	41,02	16,5	28,16
	Homotopy	4.549,69	1.769,69	3.375,16	796,01	1.789,06
	Lasso	60.446,83	24.548,87	42.517,01	9.134,15	16.143,63
	Elastic net	60.662,42	24.322,03	42.462,46	9.623,69	17.350,25
	SSF	24.738,88	10.268,87	23.033,07	6.947,7	14.542,06
	PCD	24.758,53	10.230,74	21.723,47	6.939,45	14.283,71

Çizelge 6.5, 6.6, 6.7 ve 6.8'de tüm algoritmalar için eğitim aşaması süreleri verilmiştir. Tüm süreler saniyedir. Katsayı öğrenme algoritmaları farklı sözlükler üzerinde

gösterilmektedir. Bütün sözlükler aynı boyutta olduğu için algoritmaların sözlükten bağımsız benzer sürelerle sahip olduğu görülmektedir. Katsayı algoritmalarından OMP en basit algoritma olduğundan en hızlı seyrek katsayı üretimini yapmaktadır. LFD algoritmaları içerisinde HoG özellikleri oldukça hızlı çıkarılmaktadır. Rastgele Özellikler sözlüğü herhangi bir işlem yapılmadığından oldukça hızlı oluşturulmaktadır.

Çizelge 6.6. Coil 20 için sözlük öğrenme süreleri (saniye)

	SIFT	SURF	HoG	LBP	LTP
Rasgele Özellikler	17,79	8,96	22,33	4,4	7,45
K-Means	2.036,03	1.621,98	2.636,95	351,21	593,33
K-SVD	4.300,95	1.856,68	3.864,02	1.257,57	2.291,62
ODL	337,37	330,16	343,08	290,54	286,59

Çizelge 6.7. Coil 20 için LFD algoritmaları süreleri (saniye)

SIFT	SURF	HoG	LBP	LTP
106,31	14,8	12,56	1.033,86	2.178,19

Çizelge 6.8. Coil 20 için FBSR algoritması süreleri (saniye)

SIFT	SURF	HoG	LBP	LTP
8.065,8	3.351,47	6.150,45	2.115,18	3.410,71

6.2.3. Corel 1000 veri kümesi sonuçları

Coil 20 veri kümesi üzerinden yapılan testlerin benzeri Corel 1000 veri kümesi için de gerçekleştirilmiştir. Çizelge 6.9’da Rastgele Özellikler ve K-Means sözlük algoritmasının farklı havuzlama yöntemleri ile Corel 1000 veri kümesi üzerinde elde edilen test sonuçları gösterilmektedir. Sonuçlar incelendiğinde ilk olarak enbüyük havuzlama yaklaşımının en iyi sonucu verdiği görülmektedir. Rastgele özellikler için en iyi birleşim LBP ya da LTP, SSF ya da PCD ve ortalama havuzlama ya da katsayıların toplamı olarak %55 MAP elde etmiştir. Diğer taraftan K-Means için en iyi birleşim LBP, Homotopy, enbüyük havuzlama ya da katsayıların toplamı olarak %57 MAP elde etmiştir.

Çizelge 6.9. Corel 1000 üzerinde 512 boyutlu sözlükler için MAP değerleri

		Rastgele Özellikler					K-Means				
		SIFT	SURF	HoG	LBP	LTP	SIFT	SURF	HoG	LBP	LTP
Ortalama Havuzlama	OMP	0,31	0,29	0,23	0,45	0,44	0,31	0,29	0,23	0,48	0,43
	Homotopy	0,43	0,39	0,51	0,48	0,47	0,39	0,36	0,49	0,55	0,53
	Lasso	0,34	0,32	0,46	0,37	0,32	0,32	0,3	0,46	0,46	0,37
	Elastic net	0,34	0,32	0,46	0,19	0,16	0,32	0,29	0,46	0,2	0,14
	SSF	0,47	0,35	0,44	0,55	0,55	0,48	0,33	0,44	0,54	0,53
	PCD	0,47	0,35	0,44	0,55	0,55	0,47	0,33	0,44	0,54	0,53
Enbüyük Havuzlama	OMP	0,33	0,3	0,2	0,44	0,44	0,34	0,27	0,2	0,41	0,34
	Homotopy	0,36	0,34	0,49	0,41	0,41	0,43	0,4	0,52	0,57	0,5
	Lasso	0,4	0,34	0,46	0,37	0,32	0,42	0,37	0,5	0,47	0,38
	Elastic net	0,39	0,32	0,45	0,19	0,16	0,42	0,32	0,49	0,19	0,14
	SSF	0,25	0,19	0,39	0,45	0,51	0,36	0,26	0,41	0,46	0,51
	PCD	0,26	0,19	0,4	0,45	0,5	0,36	0,26	0,42	0,46	0,52
Katsayıların Toplamı	OMP	0,42	0,31	0,23	0,45	0,44	0,42	0,3	0,23	0,48	0,43
	Homotopy	0,47	0,44	0,51	0,48	0,47	0,46	0,43	0,49	0,55	0,53
	Lasso	0,42	0,36	0,46	0,37	0,32	0,43	0,32	0,46	0,46	0,37
	Elastic net	0,42	0,35	0,46	0,19	0,16	0,43	0,3	0,46	0,2	0,14
	SSF	0,49	0,4	0,44	0,55	0,55	0,5	0,39	0,44	0,54	0,53
	PCD	0,49	0,4	0,44	0,55	0,55	0,5	0,39	0,44	0,54	0,53

Çizelge 6.10, 512 boyutlu K-SVD sözlükleri için MAP değerlerini göstermektedir. Coil 20 için farklı sözlük boyutları incelendiğinde belirgin bir farklılık gözlemlenmediğinden burada 256 boyutlu sözlükler incelenmemiştir. Çizelge incelendiğinde en iyi birleşim LBP, SSF ya da PCD ve ortalama havuzlama ya da katsayıların toplamı olarak %54 MAP elde etmiştir.

CBIR çatısından elde edilen tüm sonuçlar incelendiğinde havuzlama yöntemlerinde Rastgele Özellikler ve K-SVD sözlükleri için Coil 20 veri kümesinde olduğu gibi ortalama havuzlama ya da katsayıların toplamı yaklaşımlarının en iyi sonuçları ürettiği görülmektedir. Katsayı öğrenme algoritmaları arasında ise bu sefer tekrarlayan-daralma tabanlı SSF ve PCD algoritmaları ile Homotopy algoritması öne çıkmıştır. Sözlük öğrenmesinde ise K-Means algoritması oldukça etkili sonuçlar vermiştir. Çizelge 6.11'de ODL algoritması kullanılarak oluşturulan CBIR modelinin sonuçları gösterilmektedir.

Model LBP ve ortalama havuzlama ya da katsayıların toplamı olarak %48 ile en iyi MAP değeri üretmiştir.

Çizelge 6.10. Corel 1000 üzerinde K-SVD sözlük için MAP değerleri

		SIFT	SURF	HoG	LBP	LTP
Rastgele Özellikler	OMP	0,29	0,28	0,44	0,46	0,34
	Homotopy	0,39	0,36	0,48	0,52	0,41
	Lasso	0,32	0,3	0,46	0,47	0,35
	Elastic net	0,32	0,29	0,47	0,4	0,27
	SSF	0,48	0,33	0,44	0,54	0,47
	PCD	0,48	0,33	0,44	0,54	0,47
K-Means	OMP	0,33	0,29	0,49	0,31	0,31
	Homotopy	0,42	0,4	0,52	0,45	0,44
	Lasso	0,42	0,38	0,49	0,42	0,36
	Elastic net	0,42	0,34	0,48	0,32	0,3
	SSF	0,35	0,25	0,41	0,46	0,5
	PCD	0,35	0,25	0,41	0,46	0,5
K-SVD	OMP	0,4	0,26	0,44	0,46	0,34
	Homotopy	0,46	0,42	0,48	0,52	0,41
	Lasso	0,43	0,33	0,46	0,47	0,35
	Elastic net	0,43	0,31	0,47	0,4	0,27
	SSF	0,5	0,39	0,44	0,54	0,47
	PCD	0,5	0,39	0,44	0,54	0,47

Çizelge 6.11. Corel 1000 üzerinde ODL algoritması için MAP değerleri

	Ortalama Havuzlama	Enbüyük Havuzlama	Katsayıların Toplamı
SIFT	0,34	0,33	0,43
SURF	0,34	0,31	0,38
HoG	0,41	0,41	0,41
LBP	0,48	0,4	0,48
LTP	0,39	0,41	0,39

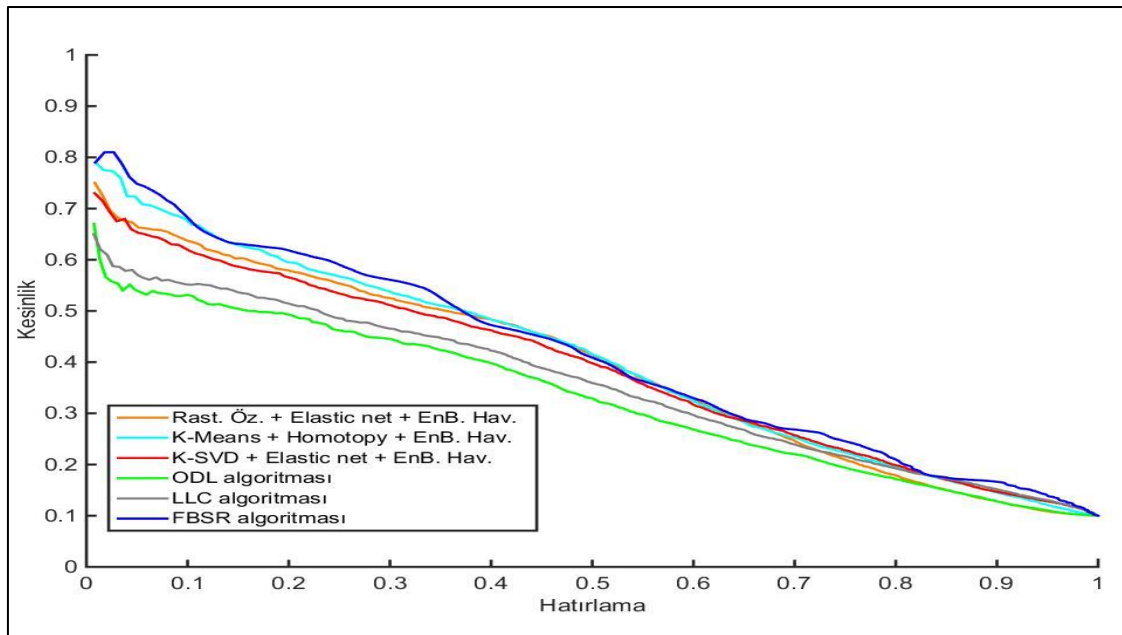
Çizelge 6.12’de tüm modellerin en iyi sonuçları gösterilmektedir. Buna göre literatürdeki yaklaşımlardan FBSR algoritması %58 ile en iyi sonucu vermiştir. Buna ek olarak CBIR çatısı yaklaşımlarında K-Means sözlük öğrenme, Homotopy katsayı öğrenme

algoritmalarını yapan ve katsayıların toplamı havuzlama yöntemini kullanan birleşim FBSR algoritmasına %57 MAP ile en yakın sonucu vermiştir. Coil 20 veri kümesinin aksine Corel 1000 veri kümesi üzerinde LLC algoritması oldukça kötü sonuçlar vermiştir.

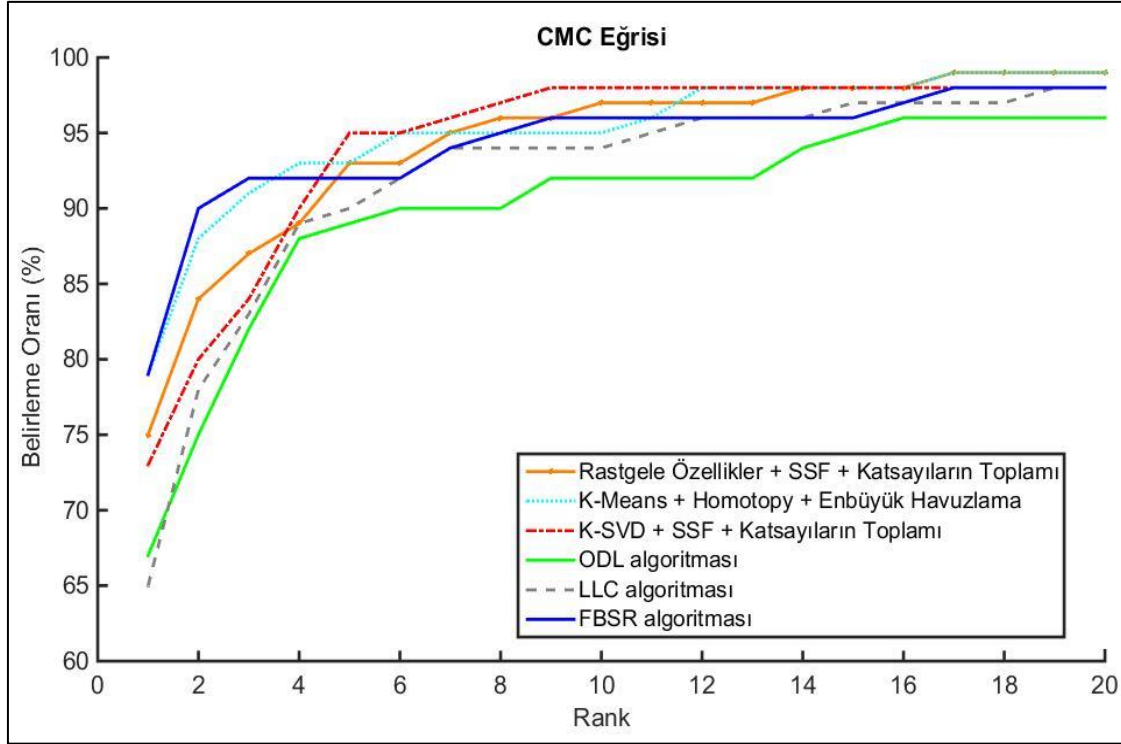
Çizelge 6.12. Corel 1000 için tüm yaklaşımların en iyi sonuçları

	SIFT	SURF	HoG	LBP	LTP
Rastgele Özellikler + SSF + Katsayıların toplamı	0,49	0,4	0,44	0,55	0,55
K-Means + Homotopy + Enbüyük Havuzlama	0,43	0,4	0,52	0,57	0,5
K-SVD + SSF + Katsayıların toplamı	0,5	0,39	0,44	0,54	0,47
Çalışma [57] – ODL algoritması	0,43	0,38	0,41	0,48	0,39
Çalışma [49] – LLC algoritması	0,25	0,2	0,5	0,35	0,28
Çalışma [32] – FBSR algoritması	0,26	0,27	0,46	0,58	0,46

Şekil 6.8 ve Şekil 6.9’da Corel 1000 veri kümesi için CBIR çatısı üzerindeki her bir sözlük için en iyi birleşimlerin ve literatürdeki çalışmaların kesinlik-hatırlama eğrileri ve CMC eğrileri verilmektedir. Bu eğriler incelendiğinde FBSR algoritması kesinlik-hatırlama eğrisinde ideale en yakın değer olmakla birlikte diğer algoritmalarla elde edilen eğriler birbirlerine oldukça yakındır. CMC eğrisinde de benzer bir sonuç elde edilmiştir.

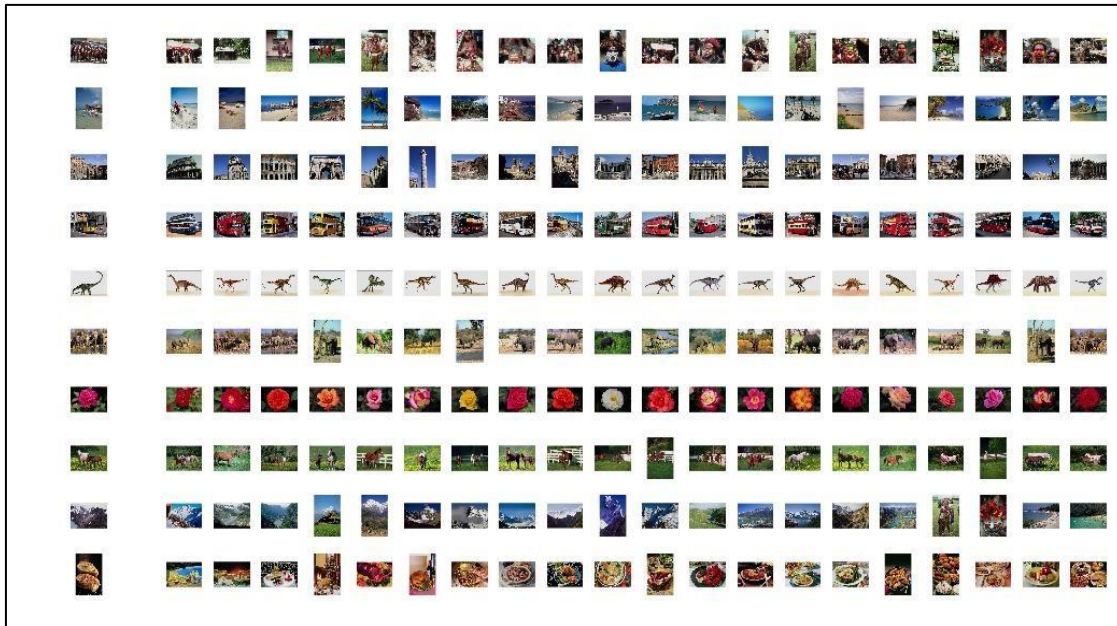


Şekil 6.8. Corel 1000 için tüm yaklaşımların kesinlik-hatırlama eğrisi



Şekil 6.9. Corel 1000 için tüm yaklaşımların CMC eğrisi

Şekil 6.10'da FBSR algoritmasının Corel 1000 veri kümesi üzerindeki her bir sınıf için elde edilen en iyi ilk 20 görüntü gösterilmektedir.



Şekil 6.10. Corel 1000 veri kümesi üzerindeki her bir sınıf için en iyi ilk 20 sonuç

Çizelge 6.13. Corel 1000 için katsayı öğrenme süreleri (saniye)

		SIFT	SURF	HoG	LBP	LTP
Rastgele Özellikler	OMP	388,57	144,38	170,74	10,21	17,85
	Homotopy	39.066,78	14.607,56	12.212,38	870,75	1.246,42
	Lasso	29.144,47	11.578,24	14.116,63	1.342,6	1513,92
	Elastic net	28.636,66	9.797,84	13.524,21	495,49	674,22
	SSF	46.797,27	18.161,22	22944,11	1103,02	1.922,55
	PCD	45.945,39	18.394,61	21.547,64	1.006,61	1.903,77
K-Means	OMP	372,25	161,05	182,43	11,45	19,17
	Homotopy	39.668,66	15.457,87	18.089,26	669,77	1.286,68
	Lasso	44.442,46	19.183,56	20.351,44	1.536,69	2.720,65
	Elastic net	42.413,3	15.117,5	19.123,61	457,05	676,2
	SSF	47.977,53	18.561,13	22.461,22	1.022,76	2.020,95
	PCD	49.098,25	18.287,69	22.336,56	1.002,06	2.000,41
K-SVD	OMP	373,92	158,47	175,8	9,98	17,98
	Homotopy	39.804,28	15.102,36	17.558,44	635,88	1.219,76
	Lasso	47.534,54	19.561,06	21.238,56	1.708,1	4.381,92
	Elastic net	44.073,76	15.754,24	19.190,58	668,44	2.753,73
	SSF	47.594,75	19.021,72	22.547,89	1.006,24	2014,5
	PCD	47.952,7	17.920,68	22.063,56	1.084,51	2.114,79

Çizelge 13, 14, 15 ve 16’da tüm algoritmaların süreleri verilmiştir. Burada Coil 20 veri kümesindeki gibi benzer sonuçlar görülmektedir. Tüm süreler saniyedir. Katsayı öğrenme algoritmaları farklı sözlükler üzerinde gösterilmektedir. Bütün sözlükler aynı boyutta olduğu için algoritmaların sözlükten bağımsız benzer sürelerle sahip olduğu görülmektedir. Katsayı algoritmalarından OMP en basit algoritma olduğundan en hızlı seyrek katsayı üretimini yapmaktadır.

Çizelge 6.14. Corel 1000 için sözlük öğrenme süreleri (saniye)

	SIFT	SURF	HoG	LBP	LTP
Rastgele Özellikler	129,08	52,69	63,9	1,97	3,41
K-Means	15.872,61	4.563,3	5.583,7	233,8	452,88
K-SVD	38.482,08	14.061,31	18.647,02	959,1	1.667,55
ODL	437,93	372,38	378,47	290,05	272,97

Çizelge 6.15. Corel 1000 için LFD algoritmaları süreleri (saniye)

SIFT	SURF	HoG	LBP	LTP
404,13	27,93	27,38	4.361,19	8.550,74

LFD algoritmaları içerisinde SURF ve HoG özellikleri oldukça hızlı çıkarılmaktadır. Rastgele Özellikler sözlüğü herhangi bir işlem yapılmadığından oldukça hızlı oluşturulmaktadır.

Çizelge 6.16. Corel 1000 için FBSR algoritması süreleri (saniye)

SIFT	SURF	HoG	LBP	LTP
33.150,49	19.751,21	23.838,81	1.597,41	2.484,06

6.3. Hızlandırılmış LFD Kümeleme Karşılaştırmalı Sonuçlar

Bu bölümde hızlandırılmış LFD kümeleme algoritması ile standart bölümlenme işlemi sonrası LFD kümeleme yaklaşımları karşılaştırılacaktır. Karşılaştırma işleminde, LFD noktalarının kümelene süreleri ve kümelene benzerlikleri başarımlı ölçütü olarak kullanılacaktır. Sonraki testlerde ise önerilen yaklaşımın erişim problemlerindeki başarımlı test edilecektir.

6.3.1. Parametreler

- LFD algoritmaları: SIFT özelliklerinin çıkarılması için Lowe tarafından hazırlanan kaynak kod kullanılmıştır. Tüm parametrelere koddaki varsayılan değerler atanmıştır. Standart Matlab fonksiyonları SURF ve HoG özelliklerini elde etmek için kullanılmıştır. SURF noktalarını elde etmek için SURFSize parametresi 128 olarak belirlenmiş ve diğer parametreler ise varsayılan değer olarak atanmıştır. HoG özelliklerini elde etmek için Block-Size=[4 4] olarak belirlenmiş ve diğer parametreler fonksiyonun varsayılan değerleri olarak belirlenmiştir. LBP8,1 ve LTP8,1 algoritmaları LBP ve LTP özelliklerini çıkarmak için kullanılmıştır. Her bir piksel 8 komşusunun bilgisi kullanılarak etiketlenmiştir.
- Bölümlenme algoritmaları: Çalışmada literatürde görüntü erişim yaklaşımlarında popüler olarak kullanılan kümeleme-tabanlı K-Means ve Mean Shift algoritmaları kullanılmıştır.

Algoritmaların görüntü piksellerinin yalnız renk bilgisini kullanan yaklaşımlarının yanı sıra hem renk hem de konum bilgisini kullanan yaklaşımları da kullanılmıştır. Her iki algoritma da 10 yinleme için çalıştırılmıştır. Hızlandırılmış kümeleme yaklaşımında pencere boyutu 3x3 ve 5x5 olarak seçilmiştir. Böylelikle farklı pencere boyutu ile standart bölümeleme algoritmaları arasındaki benzerlikler karşılaştırılmıştır.

6.3.2. Karşılaştırmalı sonuçlar

Çizelge 6.17'de Photography veri kümesi için kümeleme süreleri verilmiştir. Sonuçlar incelendiğinde önerilen yaklaşımların farklı pencere boyutlarında ve tüm LFD'ler için K-Means algoritması üzerinde daha hızlı kümeleme işlemi yaptığı gözlemlenmektedir. Mean Shift algoritmasında ise yalnızca SIFT özellikleri üzerinde Mean Shift algoritmasının piksel değerlerinin yalnız renk bilgisini kullandığı sürümü 5x5 pencere boyutuna göre daha hızlı kümeleme işlemi yapmıştır.

Çizelge 6.17. Photography veri kümesi için kümeleme süreleri (dakika)

Bölümeleme Algoritmaları	Yerel Özellik Tanımlayıcıları				
	SIFT	SURF	HoG	LBP	LTP
3x3 Pencere					
Hızlı K-Means (Renk)	32,46	8,81	14,66	3,09	1,68
Hızlı K-Means (Renk+Konum)	32,67	8,89	14,81	3,09	1,7
Hızlı Mean Shift (Renk)	3,57	1,38	1,66	0,57	0,43
Hızlı Mean Shift (Renk+Konum)	9,58	2,33	3,67	0,84	0,48
5x5 Pencere					
Hızlı K-Means (Renk)	96,7	23,67	40,18	7,72	4,05
Hızlı K-Means (Renk+Konum)	89,72	23,83	40,2	7,75	4,05
Hızlı Mean Shift (Renk)	9,03	3,11	3,87	0,99	0,62
Hızlı Mean Shift (Renk+Konum)	31,87	5,81	10,26	1,56	0,75
Standart Algoritma					
K-Means (Renk)	425,52	425,42	427,16	429,27	424,51
K-Means (Renk+Konum)	425,28	425,1	426,18	434,04	426,37
Mean Shift (Renk)	8,01	7,3	7,56	7,3	7,23
Mean Shift (Renk+Konum)	228,21	231,33	233,5	231,05	226,62

Çizelge 6.18’de Photography veri kümesi üzerinde algoritmaların kümeleme benzerlikleri verilmiştir. Sonuçlar incelendiğinde 3x3 pencere boyutu kullanan hızlandırılmış kümeleme algoritması standart kümelemeye oldukça benzer sonuçlar üretmiştir. Yalnızca Mean Shift algoritmasının piksellerin renk ve konum bilgilerini kullandığı sürümünde sonuçlar düşük çıkmıştır. Özellikler üzerinden karşılaştırıldığında ise en yüksek sonuç HoG özellikleri üzerinde elde edilmiştir. Bölümeleme algoritmaları incelendiğinde ise Mean Shift algoritmasının piksellerin renk sürümünü kullandığı sürümü en iyi sonucu vermiştir. 5x5 pencere boyutuna göre elde edilen sonuçlar incelendiğinde ise, 3x3 pencere boyutu ile elde edilen kümeleme benzerlikleri birbirlerine yakındır. Ancak genel olarak 3x3 pencere boyutu kullanan hızlandırılmış LFD kümeleme yaklaşımı, standart kümeleme yaklaşımına daha benzer kümeleme işleme gerçekleştirmiştir.

Çizelge 6.18. Photography veri kümesi için kümeleme benzerlikleri

Bölümeleme Algoritmaları	Yerel Özellik Tanımlayıcıları				
	SIFT	SURF	HoG	LBP	LTP
3x3 Pencere					
Hızlı K-Means - K-Means(Renk)	0,85	0,9	0,9	0,89	0,88
Hızlı K-Means - K-Means (Renk+Konum)	0,82	0,81	0,86	0,86	0,84
Hızlı Mean Shift – Mean Shift (Renk)	0,93	0,88	0,98	0,96	0,94
Hızlı Mean Shift – Mean Shift (Renk+Konum)	0,7	0,69	0,83	0,67	0,57
5x5 Pencere					
Hızlı K-Means - K-Means(Renk)	0,78	0,86	0,87	0,87	0,86
Hızlı K-Means - K-Means (Renk+Konum)	0,79	0,8	0,86	0,85	0,84
Hızlı Mean Shift – Mean Shift (Renk)	0,92	0,89	0,97	0,96	0,95
Hızlı Mean Shift – Mean Shift (Renk+Konum)	0,69	0,69	0,83	0,69	0,58

Çizelge 6.19’da ImageCLEF 2008 veri kümesi için kümeleme süreleri verilmiştir. Sonuçlar incelendiğinde önerilen yaklaşımların farklı pencere boyutlarında ve tüm LFD’ler için hem K-Means hem de Mean Shift algoritması üzerinde daha hızlı kümeleme işlemi yaptığı gözlemlenmektedir. Çizelge 6.20’de ImageCLEF 2008 veri kümesi üzerinde algoritmaların kümeleme benzerlikleri verilmiştir. Sonuçlar incelendiğinde 3x3 pencere boyutu kullanan hızlandırılmış kümeleme algoritması standart kümelemeye benzer

sonuçlar üretmemiştir. Yalnızca Mean Shift algoritmasının piksellerin renk bilgilerini kullandığı sürümünde sonuçlar oldukça iyi çıkmıştır. Özellikler üzerinden karşılaştırıldığında ise en yüksek sonuç HoG ve LBP özellikleri üzerinde elde edilmiştir.

Çizelge 6.19. ImageCLEF 2008 veri kümesi için bölümlenme süreleri (dakika)

Bölümlenme Algoritmaları	Yerel Özellik Tanımlayıcıları				
	SIFT	SURF	HoG	LBP	LTP
3x3 Pencere					
Hızlı K-Means (Renk)	52,77	17,52	24,26	3,37	1,71
Hızlı K-Means (Renk+Konum)	48,49	17,75	24,7	3,14	1,73
Hızlı Mean Shift (Renk)	4,62	2,02	2,15	0,51	0,39
Hızlı Mean Shift (Renk+Konum)	10,64	3,44	4,1	0,69	0,42
5x5 Pencere					
Hızlı K-Means (Renk)	145,64	47,15	65,4	7,66	4,03
Hızlı K-Means (Renk+Konum)	139,5	47,26	65,88	7,66	4,02
Hızlı Mean Shift (Renk)	12,54	4,93	5,22	0,84	0,54
Hızlı Mean Shift (Renk+Konum)	37,63	8,95	10,91	1,21	0,62
Standart Algoritma					
K-Means (Renk)	705,31	705,3	712,38	713,43	705,06
K-Means (Renk+Konum)	710,53	714,74	732,03	722,61	709,41
Mean Shift (Renk)	16,74	16,45	16,51	16,14	15,92
Mean Shift (Renk+Konum)	202,98	203,99	201,78	201,45	200,91

5x5 pencere boyutuna göre elde edilen sonuçlar incelendiğinde ise, 3x3 pencere boyutu ile elde edilen kümeleme benzerlikleri birbirlerine yakındır. Ancak genel olarak 3x3 pencere boyutu kullanan hızlandırılmış LFD kümeleme yaklaşımı, standart kümeleme yaklaşımına daha benzer kümeleme işleme gerçekleştirmiştir. Yine 5x5 pencere boyutu kullanan hızlandırılmış LFD kümeleme yaklaşımı da 3x3 pencere boyutu kullanan sürümünde olduğu gibi en iyi sonucu Mean Shift bölümlenme algoritması ile elde edilmiştir. Her iki veri kümesi üzerinden elde edilen sonuçlar incelendiğinde Photography veri kümesi üzerinde ki benzerliklerin ImageCLEF 2008 veri kümesine göre genel olarak daha yüksek olduğu gözlemlenmiştir. Her iki veri kümesi için farklı pencere boyutlarının benzerlik üzerinde çok büyük bir farklılık yaratmadığı gözlemlenmiştir. Özellikle Mean Shift

algoritmasının piksellerin renk bilgilerini kullandığı yaklaşım tüm LFD özellikleri üzerinde yüksek benzerlik oranı elde etmiştir. Özellikle bu algoritma ile önerilen yöntem LFD özelliklerinin kümelenebilirliğinde kullanılabilir.

Çizelge 6.20. ImageCLEF 2008 veri kümesi için kümeleme benzerlikleri

Bölümleme Algoritmaları	Yerel Özellik Tanımlayıcıları				
	SIFT	SURF	HoG	LBP	LTP
3x3 Pencere					
Hızlı K-Means - K-Means(Renk)	0,57	0,65	0,68	0,70	0,70
Hızlı K-Means - K-Means (Renk+Konum)	0,60	0,63	0,71	0,70	0,69
Hızlı Mean Shift – Mean Shift (Renk)	0,84	0,83	0,94	0,92	0,88
Hızlı Mean Shift – Mean Shift (Renk+Konum)	0,68	0,7	0,86	0,71	0,61
5x5 Pencere					
Hızlı K-Means - K-Means(Renk)	0,44	0,58	0,64	0,66	0,67
Hızlı K-Means - K-Means (Renk+Konum)	0,55	0,6	0,69	0,69	0,67
Hızlı Mean Shift – Mean Shift (Renk)	0,83	0,82	0,93	0,92	0,88
Hızlı Mean Shift – Mean Shift (Renk+Konum)	0,66	0,68	0,86	0,72	0,61

6.4. Çevrimdışı RBIR Modeli Sonuçları

Önerilen çevrimdışı RBIR modeli üç farklı LFD noktası kullanılarak test edildikten sonra elde edilen en iyi LFD çevrimiçi modelde kullanılmıştır.

6.4.1. Parametreler

Önerilen çatının her bir adımında kullanılan algoritmalar ve bu algoritmaların parametreleri aşağıdaki gibidir:

- LFD algoritmaları: HoG özelliklerini elde etmek için Block-Size=[4 4] olarak belirlenmiş ve diğer parametreler fonksiyonun varsayılan değerleri olarak belirlenmiştir. LBP_{8,1} ve LTP_{8,1} algoritmaları LBP ve LTP özelliklerini çıkarmak için kullanılmıştır.
- Bölümleme algoritmaları: Her iki algoritma da 10 yinleme için çalıştırılmıştır. Pencere boyutu 3x3 olarak seçilmiştir.

- Sözlük öğrenme: Sözlük boyutu 512 olarak belirlenmiştir. K-Means algoritması 100 yineleme için eğitilmiştir.
- Katsayı öğrenme: SSF algoritması 10 yineleme için eğitilmiş ve λ parametresi 0.01'dir.
- Havuzlama: Üç havuzlama yöntemi kullanılmıştır. Bu yöntemler ortalama havuzlama, enbüyük havuzlama ve katsayıların toplamıdır.

6.4.2. Karşılaştırmalı sonuçlar

Çizelge 6.21. HoG için çevrimdışı RBIR sonuçları

		ImageCLEF 2008			Photography		
		GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
Ortalama Havuzlama	Hızlı K-Means (Renk)	0,7	0,66	64,13	0,26	0,25	47,93
	Hızlı K-Means (Renk+Konum)	0,7	0,66	64,58	0,27	0,26	47,83
	Hızlı Mean Shift (Renk)	0,7	0,66	21,51	0,26	0,25	26,79
	Hızlı Mean Shift (Renk+Konum)	0,71	0,67	31,18	0,26	0,25	38,89
	K-Means (Renk)	0,7	0,66	838,84	0,26	0,25	529,37
	K-Means (Renk+Konum)	0,7	0,66	853,07	0,27	0,25	524,86
	Mean Shift (Renk)	0,71	0,66	37,26	0,26	0,25	33,86
	Mean Shift (Renk+Konum)	0,71	0,67	251,49	0,25	0,24	288,96
Enbüyük Havuzlama	Hızlı K-Means (Renk)	0,67	0,63	66,54	0,25	0,24	54,59
	Hızlı K-Means (Renk+Konum)	0,67	0,63	67,14	0,25	0,24	54,6
	Hızlı Mean Shift (Renk)	0,7	0,66	27,4	0,26	0,25	36,64
	Hızlı Mean Shift (Renk+Konum)	0,71	0,66	36,75	0,25	0,24	49,08
	K-Means (Renk)	0,66	0,62	755,83	0,25	0,24	467,19
	K-Means (Renk+Konum)	0,67	0,63	775,54	0,25	0,24	466,1
	Mean Shift (Renk)	0,7	0,66	41,74	0,26	0,25	42,93
	Mean Shift (Renk+Konum)	0,71	0,67	235,22	0,23	0,23	278,08
Katsayıların Toplamı	Hızlı K-Means (Renk)	0,66	0,62	59,79	0,24	0,23	43,86
	Hızlı K-Means (Renk+Konum)	0,66	0,63	60,34	0,24	0,23	43,7
	Hızlı Mean Shift (Renk)	0,69	0,65	20,91	0,25	0,24	26,24
	Hızlı Mean Shift (Renk+Konum)	0,7	0,66	29,99	0,25	0,24	37,69
	K-Means (Renk)	0,66	0,62	749,04	0,24	0,23	456,29
	K-Means (Renk+Konum)	0,66	0,62	768,78	0,24	0,23	455,2
	Mean Shift (Renk)	0,69	0,65	35,31	0,26	0,25	32,55
	Mean Shift (Renk+Konum)	0,7	0,66	228,34	0,23	0,23	266,74

Çizelge 6.22. HoG için çevrimdışı RBIR süreleri (dakika)

		Hızlı K-Means (Renk)	Hızlı K-Means (Renk+Konum)	Hızlı Mean Shift (Renk)	Hızlı Mean Shift (Renk+Konum)	K-Means (Renk)	K-Means (Renk+Konum)	Mean Shift (Renk)	Mean Shift (Renk+Konum)
ImageCLEF 2008	LFD	1,06	1,06	1,06	1,06	1,06	1,06	1,06	1,06
	Bölümleme	24,26	24,7	2,15	4,1	712,38	732,03	16,51	201,78
	Sözlük Öğrenme	12,38	12,55	3,65	7,18	13,19	13,53	3,64	7,83
	Katsayı Öğrenme	11,02	10,98	3,7	6,94	11,31	11,08	3,74	6,96
	Test	15,42	15,3	10,96	11,9	100,91	95,38	12,32	33,87
	Toplam	64,13	64,58	21,51	31,18	838,84	853,07	37,26	251,49
Photography	LFD	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92
	Bölümleme	14,66	14,81	1,66	3,67	427,16	426,18	7,56	233,5
	Sözlük Öğrenme	5,76	5,68	3,22	8,39	5,73	5,65	3,59	7,54
	Katsayı Öğrenme	5,35	5,35	3,71	7,45	5,42	5,42	3,72	7,46
	Test	21,24	21,07	17,28	18,47	90,14	86,7	18,08	39,54
	Toplam	47,93	47,83	26,79	38,89	529,37	524,86	33,86	288,96

Çizelge 6.21 ve 6.22’de sırası ile HoG özellikleri için önerilen yöntemin farklı bölümleme ve havuzlama yaklaşımları için iki veri kümesi üzerindeki erişim başarımları ve modellerin oluşturulma ve modeller üzerinde sorgulama süreleri gösterilmektedir.

Çizelge 6.21 incelendiğinde erişim başarımları olarak ortalama havuzlama yaklaşımının diğer havuzlama yaklaşımlarına göre daha iyi sonuçlar elde ettiği görülmektedir. Bölümleme algoritmalarına bakıldığında hızlı kümeleme yaklaşımları ile standart yaklaşımların başarımları arasında belirgin bir farklılık yoktur. Çizelge 6.22’deki modellerin oluşturulmasında her bir adım için gerekli süreler incelendiğinde özelliklerin kümelmesi işlemi modelin oluşturma maliyetinde ciddi bir yük oluşturmaktadır.

Çizelge 6.23 ve 6.24'te sırası ile [8 8] bloktan oluşan LBP özellikleri için önerilen yöntemin farklı bölümlere ve havuzlama yaklaşımları için iki veri kümesi üzerindeki erişim başarımları ve modelin oluşturulma süreleri gösterilmektedir.

Çizelge 6.23. [8 8] LBP için çevrimdışı RBIR sonuçları

		ImageCLEF 2008			Photography		
		GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
Ortalama Havuzlama	Hızlı K-Means (Renk)	0,72	0,68	242,49	0,32	0,3	212,57
	Hızlı K-Means (Renk+Konum)	0,72	0,68	242,82	0,33	0,3	194,19
	Hızlı Mean Shift (Renk)	0,72	0,68	227,53	0,33	0,3	187,68
	Hızlı Mean Shift (Renk+Konum)	0,72	0,68	238,35	0,31	0,29	200,57
	K-Means (Renk)	0,72	0,68	1034,64	0,32	0,3	687,46
	K-Means (Renk+Konum)	0,72	0,68	1045,36	0,32	0,3	693,15
	Mean Shift (Renk)	0,72	0,68	244,4	0,33	0,3	194,88
	Mean Shift (Renk+Konum)	0,72	0,68	456,14	0,32	0,3	445,59
Enbüyük Havuzlama	Hızlı K-Means (Renk)	0,65	0,61	247,91	0,3	0,27	202,82
	Hızlı K-Means (Renk+Konum)	0,65	0,61	247,92	0,31	0,29	202,95
	Hızlı Mean Shift (Renk)	0,71	0,67	233,6	0,32	0,29	197,8
	Hızlı Mean Shift (Renk+Konum)	0,71	0,66	243,97	0,28	0,26	210,98
	K-Means (Renk)	0,65	0,61	958,03	0,3	0,28	629,43
	K-Means (Renk+Konum)	0,65	0,61	967,5	0,31	0,28	634,16
	Mean Shift (Renk)	0,71	0,66	248,94	0,32	0,29	204,09
	Mean Shift (Renk+Konum)	0,71	0,67	439,79	0,29	0,27	434,77
Katsayıların Toplamı	Hızlı K-Means (Renk)	0,64	0,6	241,17	0,3	0,28	191,86
	Hızlı K-Means (Renk+Konum)	0,64	0,61	241,17	0,3	0,28	192,05
	Hızlı Mean Shift (Renk)	0,7	0,66	227,19	0,31	0,28	187,29
	Hızlı Mean Shift (Renk+Konum)	0,71	0,67	237,19	0,29	0,27	199,25
	K-Means (Renk)	0,64	0,6	951,07	0,3	0,28	618,3
	K-Means (Renk+Konum)	0,65	0,61	960,6	0,3	0,28	623,07
	Mean Shift (Renk)	0,69	0,65	242,41	0,3	0,28	193,59
	Mean Shift (Renk+Konum)	0,71	0,67	432,9	0,28	0,26	423,18

Çizelge 6.23 incelendiğinde erişim başarımları olarak ortalama havuzlama yaklaşımının diğer havuzlama yaklaşımlarına göre daha iyi sonuçlar elde ettiği görülmektedir. Bölümlere algoritmalarına bakıldığında hızlı kümeleme yaklaşımları ile standart yaklaşımların başarımları arasında belirgin bir farklılık yoktur.

Ancak Çizelge 6.24'teki modellerin oluşturulmasında her bir adım için gerekli süreler incelendiğinde özelliklerin kümelmesi işlemi modelin oluşturma maliyetinde ciddi bir yük oluşturmaktadır. Elde edilen sonuçlar genel olarak HoG özellikleri kullanılarak elde edilen sonuçlara göre benzerdir. Ancak LBP özellikleri ImageClef 2008 için %72 GMAP ve Photography için %33 GMAP ile en yüksek erişim başarımını elde etmiştir.

Çizelge 6.24. [8 8] LBP için çevrimdışı RBIR süreleri (dakika)

		Hızlı K-Means (Renk)	Hızlı K-Means (Renk+Konum)	Hızlı Mean Shift (Renk)	Hızlı Mean Shift (Renk+Konum)	K-Means (Renk)	K-Means (Renk+Konum)	Mean Shift (Renk)	Mean Shift (Renk+Konum)
ImageCLEF 2008	LFD	211,06	211,06	211,06	211,06	211,06	211,06	211,06	211,06
	Bölümleme	3,37	3,14	0,51	0,69	713,43	722,61	16,14	201,45
	Sözlük Öğrenme	5	5,19	1,39	4,4	4,79	5,1	1,32	2,89
	Katsayı Öğrenme	10,88	10,92	3,83	10,23	10,94	10,98	3,6	6,94
	Test	12,19	12,51	10,74	11,98	94,43	95,61	12,27	33,81
	Toplam	242,49	242,82	227,53	238,35	1034,64	1045,36	244,4	456,14
Photography	LFD	164,72	164,72	164,72	164,72	164,72	164,72	164,72	164,72
	Bölümleme	3,09	3,09	0,57	0,84	429,27	434,04	7,3	231,05
	Sözlük Öğrenme	1,9	2,12	1,49	5,2	2,13	2,14	1,29	2,97
	Katsayı Öğrenme	5,28	5,29	3,79	11,02	5,3	5,29	3,61	7,29
	Test	37,57	18,97	17,11	18,78	86,04	86,96	17,96	39,55
	Toplam	212,57	194,19	187,68	200,57	687,46	693,15	194,88	445,59

Çizelge 6.25 ve 6.26'de sırası ile [8 8] bloktan oluşan LTP özellikleri için önerilen yöntemin farklı bölümleme ve havuzlama yaklaşımları için iki veri kümesi üzerindeki erişim başarımları ve modelin oluşturulma süreleri gösterilmektedir. Çizelge 6.25 incelendiğinde erişim başarımı olarak ortalama havuzlama yaklaşımının diğer havuzlama yaklaşımlarına göre daha iyi sonuçlar elde ettiği görülmektedir. Bölümleme algoritmalarına bakıldığında hızlı kümeleme yaklaşımları ile standart yaklaşımların

başarımı arasında belirgin bir farklılık yoktur. Çizelge 6.26'daki modellerin oluşturulmasında her bir adım için gerekli süreler incelendiğinde özelliklerin kümelmesi işlemi modelin oluşturma maliyetinde ciddi bir yük oluşturmaktadır. Elde edilen sonuçlar genel olarak HoG ve LBP özellikleri kullanılarak elde edilen sonuçlara göre benzerdir. Ancak LBP özellikleri ImageClef 2008 için %72 GMAP ve Photography için %33 GMAP ile en yüksek erişim başarımını elde etmiştir.

Çizelge 6.25. LTP için çevrimdışı RBIR sonuçları

		ImageCLEF 2008			Photography		
		GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
Ortalama Havuzlama	Hızlı K-Means (Renk)	0,7	0,66	487,98	0,28	0,26	380,18
	Hızlı K-Means (Renk+Konum)	0,7	0,66	443,45	0,29	0,26	343,28
	Hızlı Mean Shift (Renk)	0,7	0,66	432,03	0,29	0,27	339,78
	Hızlı Mean Shift (Renk+Konum)	0,7	0,66	444,31	0,29	0,27	353,91
	K-Means (Renk)	0,7	0,66	1228,43	0,28	0,26	833,51
	K-Means (Renk+Konum)	0,7	0,66	1234,65	0,29	0,27	836,18
	Mean Shift (Renk)	0,7	0,66	447,29	0,28	0,27	345,72
	Mean Shift (Renk+Konum)	0,7	0,66	659,35	0,28	0,26	592,02
Enbüyük Havuzlama	Hızlı K-Means (Renk)	0,66	0,62	448,5	0,26	0,24	352,51
	Hızlı K-Means (Renk+Konum)	0,67	0,63	448,73	0,28	0,25	352,27
	Hızlı Mean Shift (Renk)	0,69	0,65	438,51	0,28	0,26	350,53
	Hızlı Mean Shift (Renk+Konum)	0,69	0,65	449,79	0,28	0,26	364,14
	K-Means (Renk)	0,66	0,62	1151,32	0,27	0,25	775,19
	K-Means (Renk+Konum)	0,67	0,63	1156,78	0,27	0,25	777,21
	Mean Shift (Renk)	0,69	0,65	451,52	0,28	0,26	354,69
	Mean Shift (Renk+Konum)	0,69	0,65	642,5	0,27	0,25	581,52
Katsayıların Toplamı	Hızlı K-Means (Renk)	0,67	0,63	441,69	0,25	0,23	341,59
	Hızlı K-Means (Renk+Konum)	0,67	0,63	441,98	0,25	0,24	341,37
	Hızlı Mean Shift (Renk)	0,69	0,64	431,37	0,27	0,25	339,31
	Hızlı Mean Shift (Renk+Konum)	0,7	0,66	443,04	0,28	0,26	353,74
	K-Means (Renk)	0,67	0,63	1144,43	0,25	0,23	764,11
	K-Means (Renk+Konum)	0,67	0,63	1149,86	0,25	0,24	765,84
	Mean Shift (Renk)	0,69	0,64	445,25	0,28	0,26	344,63
	Mean Shift (Renk+Konum)	0,7	0,66	635,75	0,26	0,24	570,07

Önerilen yöntemin tüm birleşimleri üzerinden elde edilen erişim başarımı ve maliyetleri incelendiğinde, LBP özellikleri ve ortalama havuzlama yaklaşımlarının en iyi başarımları

oranını verdikleri ve hızlandırılmış LFD kümeleme yaklaşımlarının standart bölümeleme sonrası LFD kümeleme yaklaşımları ile kıyaslandığında model maliyetini büyük ölçüde azalttığı görülmüştür.

Çizelge 6.26. LTP için çevrimdışı RBIR süreleri (dakika)

		Hızlı K-Means (Renk)	Hızlı K-Means (Renk+Konum)	Hızlı Mean Shift (Renk)	Hızlı Mean Shift (Renk+Konum)	K-Means (Renk)	K-Means (Renk+Konum)	Mean Shift (Renk)	Mean Shift (Renk+Konum)
ImageCLEF 2008	LFD	414,37	414,37	414,37	414,37	414,37	414,37	414,37	414,37
	Bölümeleme	1,71	1,73	0,39	0,42	705,06	709,41	15,92	200,91
	Sözlük Öğrenme	4,59	4,35	1,5	5,14	4,4	4,75	1,28	2,75
	Katsayı Öğrenme	10,22	10,71	4,24	12,15	9,81	10,49	3,45	6,56
	Test	57,08	12,29	11,54	12,23	94,8	95,63	12,27	34,76
	Toplam	487,98	443,45	432,03	444,31	1.228,43	1.234,65	447,29	659,35
Photography	LFD	315,7	315,7	315,7	315,7	315,7	315,7	315,7	315,7
	Bölümeleme	1,68	1,7	0,43	0,48	424,51	426,37	7,23	226,62
	Sözlük Öğrenme	2,11	1,88	1,49	5,96	1,85	1,88	1,42	2,72
	Katsayı Öğrenme	5,26	5,26	3,98	12,65	5,15	5,28	3,45	7,02
	Test	55,43	18,74	18,17	19,12	86,29	86,94	17,91	39,96
	Toplam	380,18	343,28	339,78	353,91	833,51	836,18	345,72	592,02

6.5. Çevrimiçi RBIR Modeli Sonuçları

Önerilen çevrimiçi RBIR modelinde, çevrimdışı modelde elde edilen sonuçlar göz önünde bulundurulduğunda en iyi sonucu LBP özellikleri verdiği için, ana LFD olarak LBP özellikleri kullanılmıştır. Daha sonra SURF noktaları aynı görüntülerden çıkarılmış ve iki LFD birlikte kullanılmıştır. Önerilen yaklaşımda hızlandırılmış LFD kümeleme algoritmaları ve standart bölümeleme sonrası LFD kümeleme yaklaşımlarının her biri

kullanılmıştır. Böylece önerilen hızlandırılmış LFD kümeleme yaklaşımının başarımı da test edilmiştir.

6.5.1. Parametreler

Önerilen çatının her bir adımında kullanılan algoritmalar ve bu algoritmaların parametreleri aşağıdaki gibidir:

- LFD algoritmaları: $LBP_{8,1}$ ve SURF algoritmaları LBP ve SURF özelliklerini çıkarmak için kullanılmıştır. LBP için [4 4] ve [8 8] olacak şekilde iki farklı blok boyutu kullanılmıştır. Böylelikle farklı LBP özellik sayısı ile modelin başarımı test edilmiştir. Standart Matlab fonksiyonu SURF özelliklerini elde etmek için kullanılmıştır. SURF noktalarını elde etmek için SURFSize parametresi 128 olarak belirlenmiş ve diğer parametreler ise varsayılan değer olarak atanmıştır.
- Bölümleme algoritmaları: Çalışmada literatürde görüntü erişim yaklaşımlarında popüler olarak kullanılan kümeleme-tabanlı K-Means ve Mean Shift algoritmaları kullanılmıştır. Algoritmaların görüntü piksellerinin yalnız renk bilgisini kullanan yaklaşımlarının yanı sıra hem renk hem de konum bilgisini kullanan yaklaşımları da kullanılmıştır. Her iki algoritma da 10 yineleme için çalıştırılmıştır. Hızlandırılmış kümeleme yaklaşımında pencere boyutu 3x3 olarak seçilmiştir.
- Sözlük öğrenme: Sözlük boyutu LBP sözlüğü için etiket sayısı kadar olurken SURF için K değeri 5 olarak belirlendiğinden etiket sayısının beş katı kadardır. Etiketli K-SVD algoritması 100 yineleme için eğitilmiştir.
- Katsayı öğrenme: SSF algoritması 10 yineleme için eğitilmiş ve λ parametresi 0.01 olarak belirlenmiştir.
- Havuzlama: Üç havuzlama yöntemi kullanılmıştır. Bu yöntemler ortalama havuzlama, enbüyük havuzlama ve katsayıların toplamıdır.

6.5.2. Karşılaştırmalı sonuçlar

Çizelge 6.27 ve 6.28'de sırası ile önerilen yöntemin, [8 8] bloktan oluşan LBP özellikleri ve SURF özellikleri için farklı bölümleme ve havuzlama yaklaşımları ile iki veri kümesi üzerindeki model oluşturulma süreleri ve erişim başarımları gösterilmektedir. $t=0.5$ olarak

alındığından her iki LFD özelliği üzerinden elde edilen benzerliğin genel benzerlik tespitine eşit katkısı vardır.

Çizelge 6.27. [8 8] LBP ve t=0,5 için çevrimiçi RBIR süreleri (dakika)

		Hızlı K-Means (Renk)	Hızlı K-Means (Renk+Konum)	Hızlı Mean Shift (Renk)	Hızlı Mean Shift (Renk+Konum)	K-Means (Renk)	K-Means (Renk+Konum)	Mean Shift (Renk)	Mean Shift (Renk+Konum)
ImageCLEF 2008	LFD	0,56	0,56	0,56	0,56	0,56	0,56	0,56	0,56
	Bölümleme	3,63	3,58	0,86	1,19	702,33	876,49	17,13	202,64
	Sözlük Öğrenme	7,62	20,34	44,61	106,36	27,07	6,58	40,02	72,77
	Katsayı Öğrenme	31,87	24,01	8,32	27,82	29,83	25,33	7,48	14,46
	Test	18,32	18,16	13,51	17,82	95,2	127,12	14,5	37,33
	Toplam	62	66,65	67,87	153,76	854,99	1.036,07	79,7	327,76
Photography	LFD	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49
	Bölümleme	3,61	3,42	0,8	1,21	439,74	486,19	7,51	227,77
	Sözlük Öğrenme	168,67	159,8	121,57	313,56	164,16	181	100,8	183,36
	Katsayı Öğrenme	48,34	47,64	33,7	101	48,53	53,69	28,49	58,18
	Test	44,63	43,84	36,79	70,72	103,65	104,21	35,69	73,7
	Toplam	265,74	255,19	193,35	486,98	756,57	825,58	172,98	543,49

Çizelge 6.27 incelendiğinde çevrimdışı RBIR modelinde olduğu gibi yine hızlandırılmış LFD kümeleme yaklaşımının standart yöntemle göre daha başarılı olduğu ve bölümleme adımının modelin oluşturulması maliyetini kritik ölçüde belirlediği gözlenmiştir.

Çizelge 6.28 incelendiğinde ImageCLEF 2008 veri kümesi için çevrimdışı RBIR modelinden farklı olarak hızlandırılmış LFD kümeleme yaklaşımı ile oluşturulan modeller, standart kümeleme yaklaşımına göre daha iyi erişim başarımı elde etmiştir. Photography veri kümesi üzerinde ise her iki kümeleme yaklaşımı da benzer sonuçlar üretmiştir. Ortalama havuzlama yaklaşımı diğer havuzlama yaklaşımlarına göre daha yüksek başarı

oranı vermektedir. Hızlandırılmış LFD kümeleme yaklaşımlarından K-Means bölümlenme algoritmasının renk ve konum bilgilerini kullanan sürümü ImageCLEF 2008 veri kümesi için %74 MAP değeri ile en iyi sonucu elde ederken. Photography veri kümesi için Mean Shift bölümlenme yaklaşımının yalnız piksellerin renk bilgisini kullanan sürümünün hızlandırılmış kümeleme ve standart kümeleme yaklaşımı dışındaki diğer tüm kümeleme yaklaşımları %32 MAP değeri elde etmiştir.

Çizelge 6.28. [8 8] LBP ve $t=0,5$ için çevrimiçi RBIR sonuçları

		ImageCLEF 2008			Photography		
		GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
Ortalama Havuzlama	Hızlı K-Means (Renk)	0,72	0,68	62	0,32	0,3	265,74
	Hızlı K-Means (Renk+Konum)	0,74	0,7	66,65	0,32	0,3	255,19
	Hızlı Mean Shift (Renk)	0,68	0,64	67,87	0,3	0,28	193,35
	Hızlı Mean Shift (Renk+Konum)	0,7	0,66	153,76	0,32	0,29	486,98
	K-Means (Renk)	0,73	0,69	854,99	0,32	0,3	756,57
	K-Means (Renk+Konum)	0,71	0,67	1.036,07	0,32	0,3	825,58
	Mean Shift (Renk)	0,68	0,64	79,7	0,3	0,29	172,98
	Mean Shift (Renk+Konum)	0,7	0,65	327,76	0,32	0,29	543,49
Enbüyük Havuzlama	Hızlı K-Means (Renk)	0,63	0,59	62	0,27	0,25	265,74
	Hızlı K-Means (Renk+Konum)	0,64	0,59	66,65	0,27	0,25	255,19
	Hızlı Mean Shift (Renk)	0,67	0,63	67,87	0,28	0,26	193,35
	Hızlı Mean Shift (Renk+Konum)	0,69	0,65	153,7	0,28	0,26	486,98
	K-Means (Renk)	0,63	0,58	854,99	0,27	0,26	756,57
	K-Means (Renk+Konum)	0,63	0,58	1.036,07	0,27	0,25	825,58
	Mean Shift (Renk)	0,67	0,62	79,7	0,29	0,27	172,98
	Mean Shift (Renk+Konum)	0,68	0,64	327,76	0,28	0,26	543,49
Katsayıların Toplamı	Hızlı K-Means (Renk)	0,63	0,58	62	0,27	0,25	265,74
	Hızlı K-Means (Renk+Konum)	0,63	0,58	66,65	0,27	0,25	255,19
	Hızlı Mean Shift (Renk)	0,67	0,63	67,87	0,28	0,26	193,35
	Hızlı Mean Shift (Renk+Konum)	0,7	0,65	153,7	0,3	0,28	486,98
	K-Means (Renk)	0,62	0,58	854,99	0,27	0,25	756,57
	K-Means (Renk+Konum)	0,62	0,57	1.036,07	0,27	0,25	825,58
	Mean Shift (Renk)	0,67	0,62	79,7	0,29	0,27	172,98
	Mean Shift (Renk+Konum)	0,69	0,64	327,76	0,29	0,27	543,49

Çizelge 6.29 ve 6.30'da sırası ile [8 8] bloktan oluşan LBP özellikleri ve SURF özellikleri için önerilen yöntemin farklı bölümlenme ve havuzlama yaklaşımları için iki veri kümesi

üzerindeki modelin oluşturulma süreleri ve erişim başarımları gösterilmektedir. $t=0.5$ olarak alındığından her iki LFD özelliği üzerinden elde edilen benzerliğin genel benzerlik tespitine eşit katkısı vardır.

Çizelge 6.29. [4 4] LBP ve $t=0,5$ için çevrimiçi RBIR süreleri (dakika)

		Hızlı K-Means (Renk)	Hızlı K-Means (Renk+Konum)	Hızlı Mean Shift (Renk)	Hızlı Mean Shift (Renk+Konum)	K-Means (Renk)	K-Means (Renk+Konum)	Mean Shift (Renk)	Mean Shift (Renk+Konum)
ImageCLEF 2008	LFD	0,53	0,53	0,53	0,53	0,53	0,53	0,53	0,53
	Bölümlleme	1,96	2,06	0,96	1,13	953,5	979,06	18,91	223,76
	Sözlük Öğrenme	7,79	44,36	62,69	26,66	47,31	7,4	49,03	27,81
	Katsayı Öğrenme	27,12	30,37	11,7	33,36	31,22	38,3	9	17,15
	Test	55,33	22,48	16,92	21,24	109,74	107,53	16,24	29,33
	Toplam	92,73	99,81	92,81	82,92	1142,31	1132,83	93,71	298,59
Photography	LFD	0,43	0,43	0,43	0,43	0,43	0,43	0,43	0,43
	Bölümlleme	1,71	1,79	0,85	1	601,21	615,66	8,38	247,84
	Sözlük Öğrenme	184,85	186,43	149,84	414,33	236,28	240,9	123,21	220,69
	Katsayı Öğrenme	52,04	54,46	38,65	128,83	69,2	68,33	33,38	69,19
	Test	85,47	57,17	46,9	99,81	95,69	95,51	40,4	165,82
	Toplam	324,5	300,28	236,67	644,39	1002,81	1020,83	205,8	703,97

Çizelge 6.29 incelendiğinde yine hızlandırılmış LFD kümeleme yaklaşımının standart yöntemle göre daha başarılı olduğu ve bölümlleme adımının modelin oluşturulması maliyetini kritik ölçüde belirlediği gözlemlenmiştir. Buna ek olarak Çizelge 6.30 incelendiğinde ImageCLEF 2008 veri kümesi için çevrimdışı RBIR modelinden farklı olarak hızlandırılmış LFD kümeleme yaklaşımı ile oluşturulan modeller standart kümeleme yaklaşımına göre daha iyi erişim başarımları elde etmiştir. Photography veri kümesi üzerinde ise her iki kümeleme yaklaşımı da benzer sonuçlar üretmiştir. Ortalama

havuzlama yaklaşımı diğer havuzlama yaklaşımlarına göre daha yüksek başarı oranı vermektedir.

Çizelge 6.30. [4 4] LBP ve $t=0,5$ için çevrimiçi RBIR sonuçları

		ImageCLEF 2008			Photography		
		GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
Ortalama Havuzlama	Hızlı K-Means (Renk)	0,73	0,69	92,73	0,33	0,3	324,5
	Hızlı K-Means (Renk+Konum)	0,74	0,69	99,81	0,33	0,31	300,28
	Hızlı Mean Shift (Renk)	0,69	0,65	92,81	0,29	0,27	236,67
	Hızlı Mean Shift (Renk+Konum)	0,7	0,66	82,92	0,32	0,29	644,39
	K-Means (Renk)	0,72	0,67	1.142,31	0,33	0,31	1002,81
	K-Means (Renk+Konum)	0,72	0,68	1.132,83	0,33	0,31	1020,83
	Mean Shift (Renk)	0,7	0,66	93,71	0,29	0,27	205,8
	Mean Shift (Renk+Konum)	0,69	0,64	298,59	0,29	0,27	703,97
Enbüyük Havuzlama	Hızlı K-Means (Renk)	0,63	0,59	92,73	0,25	0,23	324,5
	Hızlı K-Means (Renk+Konum)	0,63	0,59	99,81	0,27	0,25	300,28
	Hızlı Mean Shift (Renk)	0,67	0,63	92,81	0,28	0,26	236,67
	Hızlı Mean Shift (Renk+Konum)	0,68	0,64	82,92	0,28	0,26	644,39
	K-Means (Renk)	0,63	0,59	1.142,31	0,26	0,24	1002,81
	K-Means (Renk+Konum)	0,66	0,61	1.132,83	0,26	0,24	1020,83
	Mean Shift (Renk)	0,68	0,63	93,71	0,28	0,26	205,8
	Mean Shift (Renk+Konum)	0,67	0,63	298,59	0,27	0,25	703,97
Katsayıların Toplamı	Hızlı K-Means (Renk)	0,63	0,58	92,73	0,26	0,24	324,5
	Hızlı K-Means (Renk+Konum)	0,63	0,58	99,81	0,27	0,25	300,28
	Hızlı Mean Shift (Renk)	0,67	0,62	92,81	0,27	0,25	236,67
	Hızlı Mean Shift (Renk+Konum)	0,7	0,66	82,92	0,3	0,28	644,39
	K-Means (Renk)	0,62	0,58	1.142,31	0,26	0,24	1002,81
	K-Means (Renk+Konum)	0,64	0,6	1.132,83	0,26	0,24	1020,83
	Mean Shift (Renk)	0,67	0,62	93,71	0,28	0,26	205,8
	Mean Shift (Renk+Konum)	0,68	0,63	298,59	0,27	0,25	703,97

Hızlandırılmış LFD kümeleme yaklaşımlarından K-Means bölümlenme algoritmasının renk ve konum bilgilerini kullanan sürümü ImageCLEF 2008 veri kümesi için %74 MAP değeri ile en iyi sonucu elde ederken. Photography veri kümesi için K-Means tabanlı kümeleme yaklaşımları %33 MAP değeri ile Mean Shift tabanlı kümeleme yaklaşımlarına göre daha iyi sonuçlar elde etmiştir. [4 4] bloktan oluşan LBP özelliklerinin kullanıldığı bu sürüm ile [8 8] bloktan oluşan LBP özelliklerinin kullanıldığı sürüm karşılaştırıldığında ImageCLEF

2008 veri kümesi üzerinde elde en iyi sonuç %74 MAP değeri ile her ikisi içinde aynıdır. Diğer taraftan Photography veri kümesi üzerinde [4 4] blok LBP kullanılan sürüm %33 MAP değeri ile daha yüksek erişim başarımı elde etmiştir.

Çizelge 6.31'de [4 4] bloktan oluşan LBP özellikleri ve SURF özellikleri için önerilen yöntemin farklı bölümler için iki veri kümesi üzerindeki erişim başarımları gösterilmektedir. $t=0$ ve $t=1$ olarak alındığından her iki LFD özelliği üzerinden elde edilen benzerliğin yalnız biri kullanılacak şekilde erişim işlemi gerçekleştirilmiştir. $t=1$ için yalnızca LBP özelliklerine göre elde edilen benzerlik kullanılırken $t=0$ için yalnızca SURF özelliklerine göre elde edilen benzerlik kullanılmaktadır. Daha önceki yaklaşımlarda ortalama havuzlama yöntemi başarılı olduğu için. Buradaki testlerde yalnızca bu havuzlama yöntemi kullanılmıştır.

Çizelge 6.31. [4 4] LBP ve ortalama havuzlama için çevrimiçi RBIR sonuçları

		ImageCLEF 2008			Photography		
		GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
LBP ($t=1$)	Hızlı K-Means (Renk)	0,69	0,65	58,67	0,32	0,29	297,61
	Hızlı K-Means (Renk+Konum)	0,69	0,65	103,85	0,31	0,29	311,93
	Hızlı Mean Shift (Renk)	0,69	0,65	95,97	0,31	0,29	245,13
	Hızlı Mean Shift (Renk+Konum)	0,7	0,65	86,94	0,31	0,29	645,57
	K-Means (Renk)	0,68	0,64	1.053,81	0,32	0,29	964,85
	K-Means (Renk+Konum)	0,69	0,65	1.051,71	0,31	0,3	994,13
	Mean Shift (Renk)	0,7	0,66	97,45	0,31	0,29	218,38
	Mean Shift (Renk+Konum)	0,68	0,64	291,08	0,3	0,28	613,38
SURF ($t=0$)	Hızlı K-Means (Renk)	0,75	0,7	60,32	0,23	0,22	297,62
	Hızlı K-Means (Renk+Konum)	0,75	0,71	103,88	0,23	0,22	311,47
	Hızlı Mean Shift (Renk)	0,69	0,64	96,13	0,17	0,17	244,2
	Hızlı Mean Shift (Renk+Konum)	0,68	0,63	86,87	0,19	0,18	643,69
	K-Means (Renk)	0,74	0,7	1.054,82	0,22	0,22	965,55
	K-Means (Renk+Konum)	0,75	0,71	1.051,88	0,23	0,23	993,77
	Mean Shift (Renk)	0,69	0,65	96,93	0,18	0,18	218,12
	Mean Shift (Renk+Konum)	0,68	0,63	291,04	0,18	0,18	612,22

Çizelge 6.31 incelendiğinde hem $t=1$ hem de $t=0$ durumu için yine hızlandırılmış LFD kümeleme yaklaşımının standart yöntemle göre daha hızlı olduğu ve bölümler adımının modelin oluşturulması maliyetini kritik ölçüde belirlediği gözlemlenmiştir. Buna ek olarak

erişim başarımı benzerdir. Ancak genel olarak elde edilen MAP değerleri $t=1$ için ImageCLEF 2008 veri kümesi için düşükken Photography veri kümesi için ise $t=0.5$ durumun ile daha benzer sonuçlar verdiği gözlemlenmiştir. $t=0$ için ise ImageCLEF 2008 veri kümesi için daha yüksek MAP değerleri elde edilirken, Photography veri kümesi için elde edilen MAP değerleri oldukça düşüktür. Bu nedenle her iki özelliğin de benzerliklerinin kullanıldığı bir yaklaşımın kullanımın daha iyi sonuç vereceği düşünülmektedir.

Çizelge 6.32. Karşılaştırmalı sonuçlar

	ImageCLEF 2008			Photography		
	GMAP	NCDG	Süre (dakika)	GMAP	NCDG	Süre (dakika)
Çevrimiçi RBIR	0,74	0,69	99,81	0,33	0,31	300,28
Çevrimdışı RBIR	0,72	0,68	244,40	0,33	0,30	194,88
CBIR Çatısı - SSF	0,72	0,68	4672,5	0,3	0,28	3312,95
CBIR Çatısı - PCD	0,72	0,68	4656,21	0,3	0,28	3289,46
Çalışma [32] – FBSR algoritması	0,7	0,65	5237,21	0,24	0,23	5438,63
Çalışma [30] – SCBA algoritması	0,69	0,65	8560,14	0,25	0,24	5007,79
Çalışma [40] – SBESC algoritması	0,7	0,66	4407,87	0,26	0,24	2880,89
Çalışma [45] – CBMIR algoritması	0,66	0,62	3,85	0,15	0,15	6,2

Çizelge 6.32. de önerilen yaklaşımlar ve literatürdeki yaklaşımların karşılaştırmalı sonuçları verilmiştir. Bu çalışmada önerilen yaklaşımlardan CBIR çatısından tekrarlayan-daralma algoritmalarını kullanan sürümleri, çevrimdışı ve çevrimiçi RBIR modelleri kullanılmıştır. Sonuçlar incelendiğinde önerilen yaklaşımların literatürdeki yaklaşımlara göre daha iyi sonuçlar verdiği gözlemlenmektedir. Çevrimiçi RBIR modeli hem erişim başarımı hem de model oluşturma süresi açısından bakıldığında en iyi değerleri elde etmiştir. Photography veri kümesi üzerindeki sonuçlar incelendiğinde çevrimiçi RBIR yaklaşımı, çevrimdışı RBIR yaklaşımı ile benzer GMAP değeri üretmiştir. Ancak NCDG değerlerine bakıldığında, çevrimiçi RBIR modeli daha iyi sonuç üretmiştir.

Yalnız literatürdeki çalışmalardan Srinivas ve diğerlerinin yapmış olduğu çalışma daha hızlı bir şekilde erişim işlemini gerçekleştirmektedir. Diğer taraftan bu yaklaşım erişim başarımı en kötü yaklaşımdır. Bu çalışmada LFD özellikleri kullanılmamaktadır. Buna göre LFD özelliklerinin taşıma, ölçekleme ve döndürme gibi işlemlere karşı dayanıklı

oluşu ve erişim sınıflandırma gibi bilgisayarla görme problemlerine daha etkili çözümler ürettiği için oldukça yüksek erişim başarımı elde etmektedir.





7. SONUÇ VE ÖNERİLER

Bu çalışmada ilk olarak literatürdeki LFD ve SR kullanan CBIR yaklaşımları ayrıntılı olarak incelenmiştir. Yapılan inceleme sonucunda literatürde yoğun olarak kullanılan algoritmalar belirlenmiş ve bu algoritmaların performanslarını incelemek için bir CBIR çatısı önerilmiştir. Yapılan incelemelerde SR ile oluşturulan CBIR yaklaşımlarında daha önce bölümlenme kullanılarak bir RBIR önerilmediği tespit edilmiştir. Bu nedenle çevrimdışı ve çevrimiçi çalışan iki RBIR modeli önerilmiştir. Son olarak önerilen RBIR modellerinin oluşturulma sürelerini azaltmak için hızlandırılmış LFD kümeleme yaklaşımı önerilmiştir. Önerilen her yaklaşım farklı senaryolarla test edilmiş ve elde edilen sonuçlar literatürdeki sonuçlarla karşılaştırılmıştır. Yapılan testler sonucunda bu çalışmada elde edilen bulgular şu şekildedir:

- Yapılan literatür taramasında LFD ve SR kullanan CBIR yaklaşımlarının yoğun olarak kullandığı algoritmalar özellik seçimi adımında SIFT, SURF, HoG ve LBP özellikleri, sözlük öğrenme adımında rastgele özelliklerin kullanımı, K-Means, K-SVD ve ODL algoritmaları ve katsayı öğrenme adımında OMP, Homotopy, LARS, Elastic net ve LLC algoritmalarıdır.
- Önerilen CBIR çatısında literatür taramasında yoğun olarak kullanıldığı belirlenen algoritmaların yanı sıra özellik seçimi için LTP ve katsayı öğrenme adımı için görüntü erişimi yaklaşımında daha önce kullanılmayan tekrarlayan-daralma algoritmaları olan SSF ve PCD algoritmaları da kullanılmıştır.
- CBIR çatısı sonuçlarına göre tüm LFD algoritmaları erişim problemlerin de iyi sonuçlar üretmektedir. Corel 1000 veri kümesi üzerindeki testler SIFT özelliklerinin oldukça iyi sonuçlar verdiğini göstermektedir. Ancak bu özelliklerin sayısı oldukça fazla olduğu için boyut indirgemeye ihtiyaç vardır.
- CBIR çatısı sonuçlarına göre K-Means ve K-SVD sözlük öğrenme algoritmaları oldukça başarılı sonuçlar vermiştir.
- CBIR çatısı sonuçlarına göre erişim işlemlerinde daha önce kullanılmamış olan SSF ve PCD algoritmalarından elde edilen sonuçlar dikkat çekicidir. Buna ek olarak LARS ve Elastic net algoritmaları da K-Means sözlükleri üzerinde oldukça başarılıdır.
- CBIR çatısı sonuçlarına göre havuzlama yöntemlerinde ortalama havuzlama genel olarak daha başarılı sonuçlar vermektedir.

- Hızlandırılmış LFD kümeleme yaklaşımının sonuçlarına göre özellikle Mean Shift algoritması üzerinde üretilen benzerlikler standart LFD kümeleme yaklaşımı yerine kullanılabileceğini göstermektedir.
- Önerilen RBIR modellerinden elde edilen sonuçlar incelendiğinde LFD kullanımının yüksek erişim performansı gösterdiği ancak erişim maliyetini yükselttiği görülmektedir.
- Önerilen RBIR modellerinden elde edilen sonuçlar incelendiğinde, LFD özelliklerinin kümelenmesi erişim performansını arttırmıştır. Kümeleme sırasında yapılan standart bölümlenme yaklaşımının modelin oluşturulması sırasında maliyeti etkileyen ana etken olduğu görülmüştür. Diğer taraftan önerilmiş olan hızlandırılmış LFD kümeleme yaklaşımının model oluşturulma maliyetini azalttığı ve erişim başarımını arttırdığı görülmüştür.
- Önerilen RBIR modellerinden elde edilen sonuçlar incelendiğinde, K-SVD maliyetli bir algoritmadır. Ancak önerilen etiket kısıtlı sürümü hem erişim başarımını arttırmış hem de gerekli eğitim süresini azaltmıştır.
- Önerilen tüm yaklaşımlar incelendiğinde genel olarak havuzlama yöntemlerinden ortalama havuzlamanın en iyi sonuçları ürettiği gözlemlenmiştir.

LFD ve SR tabanlı CBIR yaklaşımları için bundan sonraki çalışmalarda özellik seçimi, artımlı sözlük oluşturma, tekrarlayan-daralma algoritmaları için farklı temsil fonksiyonları ve havuzlama yöntemleri için bölge duyarlı ağırlıklandırma konularının çalışılması önerilmektedir. Buna ek olarak, Temel Bileşen Analizi (Principal Component Analysis) gibi boyut indirgeme yaklaşımları ve derin öğrenme (deep learning) ile karşılaştırmalı çalışmaları önerilmektedir.

KAYNAKLAR

1. Vassilieva, N. (2009). Content-based image retrieval methods. *Programming and Computer Software*, 35 (3), 158-180.
2. Alemu, Y., Koh, J.-B., Ikram, M. and Kim, D.-K. (2009, September). *Image Retrieval in Multimedia Databases: A Survey*. Paper presented at the Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Japan.
3. Huang, W., Gao, Y. and Chan, K. (2010). A Review of Region-Based Image Retrieval. *Journal of Signal Processing Systems*, 59 (2), 143-161.
4. Smeulders, A., Worring, M., Santini, S., Gupta, A. and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (12), 1349-1380.
5. Rehman, M., Iqbal, M., Sharif, M. and Raza, M. (2012). Content based image retrieval: Survey. *World Applied Sciences Journal*, 19 (3), 404-412.
6. Datta, R., Joshi, D., Li, J. and Wang, J. Z. (2008). Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, 40 (2), 1-60.
7. Zhang, D., Islam, M. M. and Lu, G. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, 45 (1), 346 – 362.
8. Kalaivani, A. and Chitrakal, S. (2013, July). *Challenges & approaches in multi-label image annotation*. Paper presented at the Fourth International Conference on Intelligent Computing, Communications and Networking Technologies, India.
9. Mishra, P., Sonam, M. and Vijayalakshmi, M. S. (2014). Content Based Image Retrieval Using Clustering Technique: A Survey. *International Journal of Research in Computer Engineering and Electronics*, 3 (2), 1-9.
10. Yasmin, M., Mohsin, S. and Sharif, M. (2014). Intelligent Image Retrieval Techniques: A Survey. *Journal of Applied Research and Technology*, 12, 87-103.
11. Lew, M. S. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions Multimedia Computing, Communications, and Applications*, 2 (1), 1-19.
12. Schaefer, G. (2013, September). *An introduction to content-based image retrieval*. Paper presented at the Eighth International Conference on Digital Information Management, Pakistan.
13. Dharani, T. and Aroquiaraj, I. (2013, February). *A survey on content based image retrieval*. Paper presented at International Conference on Pattern Recognition, Informatics and Mobile Engineering, India.

14. Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60 (2), 91-110.
15. Bay, H., Ess, A., Tuytelaars, T. and Gool, L. V. (2008). Speeded-Up Robust Features. *Computer Vision and Image Understanding*, 110 (3), 346 – 359.
16. Dalal, N. and Triggs, B. (2005, June). *Histograms of oriented gradients for human detection*. Paper presented at IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA.
17. Ojala, T., Pietikainen, M. and Harwood, D. (1994, October). *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions*. Paper presented at Proceedings of the 12th IAPR International Conference on Pattern Recognition, Israel.
18. Tan, X. and Triggs, B. (2010). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, 19 (6), 1635-1650.
19. Gao, Y., Chan, K. L. and Yau, W.-Y. (2007, December). *Learning in content based image retrieval - a brief review*. Paper presented at 6th International Conference on Information, Communications Signal Processing, Singapore.
20. Mairal, J., Bach, F. and Ponce, J. (2014). Sparse Modeling for Image and Vision Processing. *Foundations and Trends® in Computer Graphics and Vision*, 8 (2-3), 85-283.
21. Wright, J., Yang, A., Ganesh, A., Sastry, S. and Ma, Y. (2009). Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (2), 210-227.
22. Celik, C. and Bilge, H. S. (2017). Content based image retrieval with sparse representations and local feature descriptors: A comparative study. *Pattern Recognition*, 68, 1 - 13.
23. Suchitra, S. and Chitrakala, S. (2013, July). *A survey on scalable image indexing and searching*. Paper presented at Fourth International Conference on Computing, Communications and Networking Technologies, India.
24. Ai, L.-f., Yu, J.-q., He, Y.-f. and Guan, T. (2013). High-dimensional indexing technologies for large scale content-based image retrieval: a review. *Journal of Zhejiang University SCIENCE C*, 14 (7), 505-520.
25. Li, S., Yu, H. and Yuan, L. (2016, April). *A Novel Approach to Remote Sensing Image Retrieval with Multi-feature VP-Tree Indexing and Online Feature Selection*. Paper presented at IEEE Second International Conference on Multimedia Big Data, Taiwan.
26. Nene, S. A., Nayar, S. K. and Murase, H. (1996). Columbia Object Image Library. Department of Computer Science, Columbia University, CUCS-005-96.

27. Wang, J. Z., Li, J. and Wiederhold, G. (2001). SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (9), 947-963.
28. Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive . Technical report, Irvine, CA: University of California, Department of Information and Computer Science.
29. Deselaers, T. and Hanbury, A. (2009). The Visual Concept Detection Task in ImageCLEF 2008. In C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, G. J. F. Jones, M. Kurimo, T. Mandl, A. Pecos and V. Petras (eds.), *Evaluating Systems for Multilingual and Multimodal Information Access: 9th Workshop of the Cross-Language Evaluation Forum*. Berlin, Springer Berlin Heidelberg, 531-538.
30. Huang, C.-M., Chen, S., Cheng, M. and Wang, Y.-C. (2012, November). *A sparse-coding based approach to clothing image retrieval*. Paper presented at International Symposium on Intelligent Signal Processing and Communications Systems, Taiwan.
31. Liu, Y., Cheng, J., Xu, C. and Lu, H. (2010). Building topographic subspace model with transfer learning for sparse representation . *Neurocomputing*, 73, 1662 - 1668.
32. Kang, L.-W., Hsu, C.-Y., Chen, H.-W., Lu, C.-S., Lin, C.-Y. and Pei, S.-C. (2011). Feature-Based Sparse Representation for Image Similarity Assessment. *IEEE Transactions on Multimedia*, 13 (5), 1019-1030.
33. Zhao, Z.-Q., Glotin, H., Xie, Z., Gao, J. and Wu, X. (2012). Cooperative Sparse Representation in Two Opposite Directions for Semi-Supervised Image Annotation. *IEEE Transactions on Image Processing*, 21 (9), 4218-4231.
34. Thiagarajan, J., Ramamurthy, K., Sattigeri, P. and Spanias, A. (2012, September). *Supervised local sparse coding of sub-image features for image retrieval*. Paper presented at 19th IEEE International Conference on Image Processing, Orlando, FL.
35. Hong, C. and Zhu, J. (2013). Hypergraph-based multi-example ranking with sparse representation for transductive learning image retrieval . *Neurocomputing*, 101, 94 - 103.
36. Ge, T., Ke, Q. and Sun, J. (2013, September). *Sparse-Coded Features for Image Retrieval*. Paper presented at British Machine Vision Conference, Bristol.
37. Huang, C.-M., Wei, C.-P. and Wang, Y.-C. (2013, July). *Active learning based clothing image recommendation with implicit user preferences*. Paper presented at IEEE International Conference on Multimedia and Expo Workshops, California, USA.
38. Qiao, Q. and Beling, P. (2009, December). *Localized Content Based Image Retrieval with Self-Taught Multiple Instance Learning*. Paper presented at IEEE International Conference on In Data Mining Workshops, Florida.
39. Lin, J., Duan, L.-Y., Chen, J., Ji, R., Luo, S. and Gao, W. (2012, March). *Learning multiple codebooks for low bit rate mobile visual search*. Paper presented at IEEE International Conference on Acoustics, Speech and Signal Processing, Japan.

40. Shi, J., Jiang, Z., Feng, H. and Zhang, L. (2012, September). *SIFT-based Elastic sparse coding for image retrieval*. Paper presented at 19th IEEE International Conference on Image Processing, Orlando, FL.
41. Ji, R., Yao, H., Liu, W., Sun, X. and Tian, Q. (2012). Task-Dependent Visual-Codebook Compression. *IEEE Transactions on Image Processing*, 21 (4), 2282-2293.
42. Chen, B.-C., Chen, Y.-Y., Kuo, Y.-H. and Hsu, W. (2013). Scalable Face Image Retrieval Using Attribute-Enhanced Sparse Codewords. *IEEE Transactions on Multimedia*, 15 (5), 1163-1173.
43. Zhu, X., Zhang, L. and Huang, Z. (2014). A Sparse Embedding and Least Variance Encoding Approach to Hashing. *IEEE Transactions on Image Processing*, 23 (9), 3737-3750.
44. Wang, D., Hoi, S., He, Y., Zhu, J., Mei, T. and Luo, J. (2014). Retrieval-Based Face Annotation by Weak Label Regularized Local Coordinate Coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36 (3), 550-563.
45. Srinivas, M., Naidu, R. R., Sastry, C. and Mohan, C. K. (2015). Content based medical image retrieval using dictionary learning. *Neurocomputing*, 168, 880 – 895.
46. Mohamadzadeh, S. and Farsi, H. (2016). Content-based image retrieval system via sparse representation. *IET Computer Vision*, 10 (1), 95-102.
47. Li, Q., Han, Y. and Dang, J. (2016). Sketch4Image: a novel framework for sketch-based image retrieval based on product quantization with coding residuals. *Multimedia Tools and Applications*, 75 (5), 2419-2434.
48. Zhang, Y., Xiang, M. and Yang, B. (2017). Graph regularized nonnegative sparse coding using incoherent dictionary for approximate nearest neighbor search. *Pattern Recognition*, 70 (Supplement C), 75-88.
49. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T. and Gong, Y. (2010, June). *Locality-constrained Linear Coding for image classification*. Paper presented at IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA.
50. Aharon, M., Elad, M. and Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54 (11), 4311-4322.
51. Pati, Y., Rezaifar, R. and Krishnaprasad, P. (1993, November). *Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition*. Paper presented at Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA.
52. Osborne, M. R., Presnell, B. and Turlach, B. (1999). A New Approach to Variable Selection in Least Squares Problems. *IMA Journal of Numerical Analysis*, 20, 389-404.

53. Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32, 407-499.
54. Zou, H. and Hastie, T. (2005). Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B* 67, 301-320.
55. Thiagarajan, J. and Spanias, A. (2011, November). *Learning dictionaries for local sparse coding in image classification*. Paper presented at Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA.
56. Lee, H., Battle, A., Raina, R. and Ng, A. Y. (2006, December). Efficient sparse coding algorithms. Paper Presented at Neural Information Processing systems, Vancouver, BC.
57. Mairal, J., Bach, F., Ponce, J. and Sapiro, G. (2009, June). *Online Dictionary Learning for Sparse Coding*. Paper presented at the Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC.
58. Nanni, L., Lumini, A. and Brahmam, S. (2014). Ensemble of different local descriptors, codebook generation methods and subwindow configurations for building a reliable computer vision system . *Journal of King Saud University – Science*, 26 (2), 89 – 100.
59. Grauman, K. and Leibe, B. (2011). *Visual Object Recognition*. Morgan & Claypool Publishers, 23-39.
60. Ciocca, G., Cusano, C., Santini, S. and Schettini, R. (2011). Halfway through the semantic gap: Prosemantic features for image retrieval. *Information Sciences*, 181 (22), 4943 - 4958.
61. Kihl, O., Picard, D. and Gosselin, P.-H. (2015). A unified framework for local visual descriptors evaluation. *Pattern Recognition*, 48 (4), 1174 – 1184.
62. Cao, J., Mao, D.-h., Cai, Q., Li, H.-s. and Du, J.-p. (2013). A review of object representation based on local features. *Journal of Zhejiang University SCIENCE C*, 14 (7), 495-504.
63. Lowe, D. (1999, September). *Object recognition from local scale-invariant features*. Paper presented at the Proceedings of the Seventh IEEE International Conference on Computer Vision, Greece.
64. Bay, H., Tuytelaars, T. and Van Gool, L. (2006, May). *SURF: Speeded Up Robust Features*. Paper presented at Proceedings of the 9th European Conference on Computer Vision - Volume Part I, Austria.
65. Ojala, T., Pietikainen, M. and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7), 971-987.

66. Tan, X. and Triggs, B. (2007). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. In S. Zhou, W. Zhao, X. Tang & S. Gong (ed.), *Analysis and Modeling of Faces and Gestures*, 4778, 168-182. Springer Berlin Heidelberg.
67. Alzuabi, A., Amira, A. and Ramzan, N. (2015). Semantic content-based image retrieval: A comprehensive study. *Journal of Visual Communication and Image Representation*, 32, 20 - 54.
68. Hosseinzadeh, M. and Khoshvaght, P. (2015). A Comparative Study of Image Segmentation Algorithms. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9 (8), 1966-1971.
69. Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (8), 888-905.
70. Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (5), 603-619.
71. Mathew, S. P., Balas, V. E., Zachariah, K. and Samuel, P. (2014). A content-based image retrieval system based on polar raster edge sampling signature. *Acta Polytechnica Hungarica*, 11 (3), 25-36.
72. Manipoonchelvi, P. and Muneeswaran, K. (2015). Significant region-based image retrieval. *Signal, Image and Video Processing*, 9 (8), 1795-1804.
73. Yuvaraj, D. and Hariharan, S. (2016). Content-based image retrieval based on integrating region segmentation and colour histogram. *The International Arab Journal of Information Technology*, 13 (1A), 203-207.
74. Pasnur, P., Arifin, A. Z. and Yuniarti, A. (2016). Query Region Determination based on Region Importance Index and Relative Position for Region-based Image Retrieval. *International Journal of Technology*, 7 (4), 654-662.
75. Elad, M. (2010). *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer Publishing Company.
76. Yang, Y., Yang, Y., Huang, Z., Shen, H. T. and Nie, F. (2011, June). *Tag localization with spatial correlations and joint group sparsity*. Paper presented at IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO.
77. Boureau, Y.-L., Bach, F., LeCun, Y. and Ponce, J. (2010, June). *Learning mid-level features for recognition*. Paper presented at IEEE Conference on Computer Vision and Pattern Recognition, San Fransisco.
78. Lazebnik, S., Schmid, C. and Ponce, J. (2006, June). *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. Paper presented

- at IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York.
79. Coates, A. and Ng, A. (2011, June). *The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization*. Paper presented at Proceedings of the 28th International Conference on Machine Learning, Washington.
 80. Rubinstein, R., Bruckstein, A. and Elad, M. (2010). Dictionaries for Sparse Representation Modeling. *Proceedings of the IEEE*, 98 (6), 1045-1057.
 81. Mairal, J., Bach, F., Ponce, J. and Sapiro, G. (2010). Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research*, 11, 19-60.
 82. Bach, F., Jenatton, R., Mairal, J. and Obozinski, G. (2012). Optimization with Sparsity-Inducing Penalties. *Foundations and Trends in Machine Learning*, 4 (1), 1-106.
 83. Donoho, D. L., Tsaig, Y., Drori, I. and Starck, J. L. (2012). Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, 58 (2), 1094-1121.
 84. Zibulevsky, M. and Elad, M. (2010). L1-L2 Optimization in Signal and Image Processing. *IEEE Signal Processing Magazine*, 27 (3), 76-88.
 85. Daubechies, I., Defrise, M. and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57 (11), 1413-1457.
 86. Elad, M., Matalon, B. and Zibulevsky, M. (2007). Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. *Applied and Computational Harmonic Analysis*, 23 (3), 346 – 367.
 87. Elad, M. (2006). Why Simple Shrinkage Is Still Relevant for Redundant Representations?. *IEEE Transactions on Information Theory*, 52 (12), 5559-5569.
 88. Narkiss, G. and Zibulevsky, M. (2005). Sequential subspace optimization method for large-scale unconstrained optimization, *CCIT Report*, 559. Israel, 1-26.
 89. Moon, H. and Phillips, P. J. (2001). Computational and performance aspects of PCA-based face-recognition algorithms. *Perception* 30, (3), 303-321.
 90. Bolle, R. M., Connell, J. H., Pankanti, S., Ratha, N. K. and Senior, A. W. (2005, October). *The relation between the ROC curve and the CMC*. Paper presented at Fourth IEEE Workshop on Automatic Identification Advanced Technologies, Buffalo, NY.
 91. Mohammad Razavi, S., Taghipour-Gorjikotaie, M. and Mehrshad, N. (2017). Multimodal biometric identification system based on finger-veins using hybrid rank-

- decision-level fusion technique. *IEEJ Transactions on Electrical and Electronic Engineering*, 12 (5), 728-735.
92. Boutell, M. R., Luo, J., Shen, X. and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37 (9), 1757-1771.
 93. Zhang, M. L. and Zhou, Z. H. (2014). A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26 (8), 1819-1837.
 94. Kekalainen, J. (2005). Binary and graded relevance in IR evaluations Comparison of the effects on ranking of IR systems. *Information Processing and Management*, 41 (5), 1019 – 1033.
 95. Zhao, F., Huang, Y., Wang, L. and Tan, T. (2015, June). *Deep Semantic Ranking Based Hashing for Multi-Label Image Retrieval*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA.
 96. Robertson, S. E., Kanoulas, E. and Yilmaz, E. (2010, July). *Extending Average Precision to Graded Relevance Judgments*. Paper presented at Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Switzerland.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÇELİK, Ceyhun
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 22.02.1986, Adana
 Medeni hali : Bekar
 Telefon : 0 (544) 845 46 67
 e-mail : celik.ceyhun@gmail.com



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam ediyor
Yüksek lisans	Çukurova Üniversitesi / Bilgisayar Mühendisliği	2011
Lisans	Çukurova Üniversitesi / Bilgisayar Mühendisliği	2009
Lise	Seyhan (Ç.E.A.Ş.) Anadolu Lisesi	2004

İş Deneyimi

Yıl	Yer	Görev
2012-Halen	İçişleri Bakanlığı	Mühendis

Yabancı Dil

İngilizce

Yayınlar

1. Celik, C. and Bilge, H. S. (2017, May). *Fast image segmentation for local feature descriptors*. Paper presented at 25th Signal Processing and Communications Applications Conference, Antalya.
2. Celik, C. and Bilge, H. S. (2017). Content based image retrieval with sparse representations and local feature descriptors: A comparative study. *Pattern Recognition*, 68, 1 - 13.
3. Celik, C. and Bilge, H. S. (2015). Ağırlıklandırılmış Koşullu Karşılıklı Bilgi İle Öznitelik Seçimi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 30(4), 585-596.



4. Celik, C. and Bilge, H. S. (Değerlendirmede). Seyrek Temsil İle Bölgesel Tabanlı Görüntü Erişimi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*.

Hobiler

Yüzme, Gitar, Dans





DİZİN

A

afin · 15
 AIA · 1
 Anlamsal boşluk · 5
 Ayırılabilir Temsil Fonksiyonları ·
 29

B

Benzerlik ölçümü · 3, 33, 37, 41
 bilgisayarla görme · 2, 3, 7, 15,
 19, 24, 25, 34
 Birikimli Eşleştirme
 Karakteristiği · 46
 bölge tabanlı · 2, 5
 Bölge Tabanlı Görüntü Erişimi ·
 2
 Bölge-kısıtlı Doğrusal Kodlama ·
 9

C

CBIR · v, 1, 2, 3, 4, 5, 6, 7, 8, 9,
 10, 11, 12, 13, 14, 21, 33, 43,
 46, 47, 76, 79, 80
 CBMIR · 11, 76
 CMC · 46, 51, 57, 87
 Coil 20 · 5, 43, 44, 47, 48, 49, 50,
 51, 52, 53, 54
 Corel 1000 · 5, 13, 43, 44, 47, 53,
 54, 55, 56, 57, 58, 59, 79

Ç

çevrimdışı · iv, 11, 13, 25, 36, 38,
 39, 43, 64, 65, 66, 67, 68, 69,
 77, 79
 Çevrimdışı · 3, 36, 38, 63, 76
 Çevrimiçi · 3, 38, 69, 76
 çıkarım · 19

D

desen · 1, 2, 12, 15, 17
 DoG · 16
 döndürme · 7, 15, 16

E

Elastic net · 12, 29, 33, 47, 48,
 49, 52, 54, 55, 58, 79
 erişim · 1, 2, 3, 5, 6, 7, 11, 13, 15,
 19, 34, 47, 59, 70, 75, 79, 80

F

FBSR · 7, 50, 53, 55, 56, 59, 76

G

GIST · 8, 12, 13
 görüntü · iv, 1, 2, 3, 4, 5, 6, 7, 8,
 9, 10, 11, 15, 16, 17, 18, 19,
 20, 23, 24, 25, 33, 34, 36, 37,
 39, 41, 42, 43, 45, 46, 59, 70,
 79
 görüntü bölümlenme · 15, 18
 Görüntü bölümlenme · 2, 18
 Görüntü İle Sorgulama · 1
 Görüntü tanımlayıcıları · 2

H

hatırlama · 13, 46, 51, 56, 77
 HLSI · v, 1
 HoG · 2, 8, 14, 15, 17, 33, 47, 48,
 49, 50, 52, 53, 54, 55, 56, 58,
 59, 60, 61, 62, 63, 64, 65, 79
 homotopy · 14
 Homotopy · 12, 28, 29, 33, 47,
 48, 49, 52, 54, 55, 56, 58, 79

I

ImageCLEF 2008 · 5, 43, 45, 62,
 63, 64, 65, 66, 67, 68, 69, 71,
 72, 74, 75, 76, 77, 83

İ

İçerik Tabanlı Görüntü Erişimi ·
 iv, 1

K

katsayı öğrenme · 3, 6, 14, 24,
 26, 36, 37, 41, 52, 58, 79
 kesinlik · 46
 K-Means · 12, 14, 21, 25, 26, 33,
 34, 35, 36, 37, 48, 50, 52, 53,
 54, 55, 56, 58, 59, 60, 61, 62,
 63, 64, 65, 66, 67, 68, 69, 70,
 71, 72, 73, 74, 75, 79
 K-SVD · 12, 14, 25, 33, 39, 40,
 49, 50, 52, 53, 54, 55, 56, 58,
 70, 79, 80, 84

L

LARS · 12, 14, 28, 79
 Lasso · 24, 28, 29, 33, 47, 48, 49,
 52, 54, 55, 58
 LBP · 2, 8, 9, 12, 14, 15, 17, 18,
 33, 36, 39, 41, 42, 47, 48, 49,
 50, 52, 53, 54, 55, 56, 58, 59,
 60, 61, 62, 63, 66, 67, 70, 71,
 72, 73, 74, 75, 79
 LFD · v, 2, 5, 6, 11, 33, 34, 35,
 36, 43, 47, 53, 59, 63, 65, 67,
 69, 70, 71, 73, 75, 79, 80
 LLC · 9, 12, 31, 47, 50, 56, 79
 LLF · 1, 11, 15



LTP · 2, 15, 17, 18, 33, 47, 48,
49, 50, 52, 53, 54, 55, 56, 58,
59, 60, 61, 62, 63, 68, 69, 79

M

Mahalanobis · 3
makine öğrenmesi · 3, 23
MAP · 46, 48, 49, 50, 54, 55
Matlab · 47, 59, 70
Mean Shift · 21, 34, 36, 39, 59,
60, 61, 62, 63, 64, 65, 66, 67,
68, 69, 70, 71, 72, 73, 74, 75,
80
Mekânsal Piramit Eşleme · 8
mekânsal yerleşim · 1, 2, 15
Metin İle Sorgulama · 1

N

NCDG · 46, 64, 66, 68, 72, 74,
75, 76
Normalleştirilmiş Azaltılmış
Birikimli Kazanç · 46
NP-Hard · 24

O

ODL · 50, 53, 55, 56, 58, 79
OMP · 12, 14, 27, 28, 29, 33, 47,
48, 49, 52, 54, 55, 58, 79
Otomatik Görüntü Ek Açıklaması
· 1

Ö

Öklid · 3, 33, 37, 41

ölçekleme · 15

P

Paralel Koordinat Düşümü · 29
PCD · 29, 31, 33, 47, 48, 49, 52,
54, 55, 58, 76, 79
Photography · 5, 43, 44, 45, 60,
61, 64, 65, 66, 67, 68, 69, 71,
72, 73, 74, 75, 76, 77

Q

QBI · 1
QBME · 8
QBT · 1, 4

R

Rastgele Özellikler · 48, 50, 54,
55, 56, 58
RBIR · v, 2, 5, 6, 33, 36, 37, 38,
41, 43, 63, 64, 65, 66, 67, 68,
69, 71, 72, 73, 74, 75, 76, 77,
79, 80
renk · 1, 2, 7, 9, 12, 15, 17, 34,
36, 39, 59, 70

S

SBESC · 10, 76
SCBA · 9, 76
SESOP · 29, 30, 31
Seyrek Temsil · iv, 3, 7
seyreklik · 13, 23, 24
SIFT · 2, 7, 9, 10, 12, 14, 15, 16,
17, 33, 47, 48, 49, 50, 52, 53,

54, 55, 56, 58, 59, 60, 61, 62,
63, 79, 84

sınıflandırma · 7, 15, 19, 24, 34
Sıralı Altuzaylık Optimizasyonu ·
29
sözlük öğrenme · 3, 6, 14, 25, 26,
36, 37, 38, 39, 53, 58, 79
Sözlük öğrenme · 3, 33, 47, 64,
70
SPM · 8, 31
SR · v, 3, 5, 6, 7, 8, 9, 10, 11, 13,
14, 23, 24, 25, 26, 30, 33, 36,
43, 47, 79, 80
SSF · 29, 30, 33, 37, 41, 47, 48,
49, 52, 54, 55, 56, 58, 64, 70,
76, 79
SURF · 2, 7, 12, 14, 15, 16, 17,
33, 39, 40, 41, 42, 47, 48, 49,
50, 52, 53, 54, 55, 56, 58, 59,
60, 61, 62, 63, 70, 75, 79, 85

Ş

şekil · 1, 2, 12, 15, 19, 20

T

taşma · 7, 15

Y

Yerel Özellik Tanımlayıcıları ·
iv, 2, 15, 34, 60, 61, 62, 63
yerel tanımlayıcılar · 2, 11



GAZİ GELECEKTİR..