

Büyük İmge Veri Tabanlarında Erişim İçin Yarı Eğitimli Görsel Temsiller

Tuğçe Döngel

**YÜKSEK LİSANS TEZİ**

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Mart 2016

Semi-Supervised Visual Representations for Large Scale Image Databases

Tuğçe Döngel

**MASTER OF SCIENCE THESIS**

Department of Electrical and Electronics Engineering

March 2016

Büyük İmge Veri Tabanlarında Erişim İçin Yarı Eğitimli Görsel Temsiller

Tuğçe Döngel

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Telekomünikasyon ve Sinyal İşleme Bilim Dalında

YÜKSEK LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Doç. Dr. Hakan Çevikalp

Bu tez TÜBİTAK tarafından 113E118 no'lu proje çerçevesinde desteklenmektedir.

Mart 2016

## ONAY

Elektrik-Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öđrencisi Tuđçe Döngel'in YÜKSEK LİSANS tezi olarak hazırladıđı "Büyük İmge Veri Tabanlarında Erişim İçin Yarı Eđitmenli Görsel Temsiller" başlıklı bu çalıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiřtir.

**Danıřman** : Doç. Dr. Hakan ÇEVİKALP

**İkinci Danıřman** : -

**Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Doç. Dr. Hakan ÇEVİKALP

**Üye** : Prof. Dr. Rıfat EDİZKAN

**Üye** : Yrd. Doç. Dr. Erkut ERDEM

**Üye** : Yrd. Doç. Dr. Hasan Serhan YAVUZ

**Üye** : Yrd. Doç. Dr. Helin DUTAĞACI

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve  
..... sayılı kararıyla onaylanmıřtır.

Prof. Dr. Hürriyet ERŐAHAN  
Enstitü Müdürü

## ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Hakan Çevikalp danışmanlığında hazırlamış olduğum “Büyük İmge Veri Tabanlarında Erişim İçin Yarı Eğitimli Görsel Temsiller” başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 21/03/2016

Tuğçe Döngel

## ÖZET

İmge erişimi, bir sorgu imgesine benzeyen imgeleri birçok imge arasından bulup sıralı bir şekilde döndürme işlemidir ve yapılan birçok çalışmaya rağmen hala tam olarak çözülemeyen zorlu bir bilgisayarlı görü alanıdır. İmge erişiminde en büyük problem imgenin görsel içeriğine bakarak imgeyi, onu betimleyen doğru anlambilimsel kelimelerle (sınıflar) eşleştirebilmektir. Bu tez kapsamında bu problemi en iyi biçimde çözebilmek amacıyla çok sayıdaki etiketsiz imgeyle birlikte sınırlı sayıdaki etiketli imgeyi de kullanan etkili makine ile öğrenme tekniklerinin kullanılması amaçlanmıştır. İmge erişiminde bir diğer büyük problem veri sayısının çok fazla olduğu gerçekçi uygulamalarda işlem yükünün de çok fazla olmasıdır.

İmge erişimi için önerilen ilk yöntemlerde düşük seviye öznitelik çıkarma ve imge öznitelik vektörleri arasındaki benzerliği ölçmeye dayalı teknikler kullanılmıştır. Ancak bu şekilde imgeleri doğru bir biçimde sınıflandırmak ya da imgenin içeriğini doğru bir şekilde anlamak mümkün değildir. Anlambilimsel boşluk problemi olarak adlandırılan problemin çözümünü kolaylaştırmak için izlenecek yollardan birisi tüm imgeleri etiketlemektir. Fakat imge sayısının çok fazla olduğu durumlarda bu işlemin elle yapılması pratikte hem pahalı hem de zor bir iştir. Bu nedenle bu tür etiketleme işlemleri yarı-otomatik veya tam-otomatik algoritmalar tarafından yapılmaktadır. Bu algoritmaların kullanımı neticesinde de genellikle etiketlerin arasında hatalar ve gereksiz kelimeler bulunmaktadır. Tez kapsamında yapılan çalışmalarda ilk olarak hatalı etiketli verilerin temizlenmesi üzerinde çalışmalar yapılmış olup bu amaçla yarı-otomatik bir etiketleme aracı geliştirilmiştir. Ardından literatürdeki makine ile öğrenme teknikleri kullanılarak mevcut etiketli veriler ile birlikte etiketsiz verileri de kullanan milyonlarca veriyle çalışabilen bir imge erişim sistemi geliştirilmiştir. Bu kapsamda, mevcut etiketli veriler ve bu imgelerin görsel benzerlikleri kullanılarak sınıf hiyerarşileri belirlenmiş daha sonra hem etiketli hem de etiketsiz verileri etkili bir şekilde kullanan transdaktif öğrenme yöntemi hızlı imge erişimi için imgeler arasındaki anlambilimsel ilişkileri yansıtan gürbüz ve etkili özet kodları oluşturulmada kullanılmıştır. Yapılan deneylerde önerilen yöntemin  $k$ -en yakın komşu algoritmasının performansını önemli ölçülerde iyileştirdiği gözlenmiştir.

**Anahtar kelimeler:** Büyük veri tabanlı imge erişimi, etiketleme, transdaktif destek vektör makineleri, özetleme, yarı eğitmenli öğrenme

## SUMMARY

Image retrieval can be defined as follows: Given a query image, finding and presenting (in ordered manner) the images depicting the same place or object in a large unordered image collections. Despite the great research efforts, image retrieval is still a challenging problem. The main problem in image retrieval is to bridge the semantic gap between the visual content and a textual description of the image. This thesis proposes to bridge this gap using machine learning techniques based on the semi-supervised information (given in terms of textual description) associated to the some images. In addition to this semantic gap problem, in order to solve the image retrieval from realistic point of view, we have to provide a large image database which brings additional computational difficulties.

Early studies generally focused on low-level feature extraction and similarity measurements for image retrieval. However due to the complexity of image understanding, it is very unlikely to discriminate images correctly based on the distance measurements by using only low-level features. To solve the challenging semantic gap problem, one feasible way is to build a textual description index for all images. However making suitable textual index manually in large image databases is too costly and difficult in practice. Instead of manual image annotation, automatic or semi-automatic annotation tools were developed. Therefore, such descriptions associated to the images are generally too noisy and include errors. In this thesis we first studies were made on the cleaning of the labeled data incorrectly and a semi-automatic labeling tool has been developed. Then, using the machine learning techniques in literature, an image retrieval system that uses available labeled data together with unlabeled data and can work with millions of data was developed. In this scope, images with labels were hierarchically organized based on their visual content similarities and labels. In the next step transductive learning and modified clustering algorithms were used. As apart of this process, efficient and robust hashing methods were generated that build binary codes that preserve the semantic similarity by using both the labeled and unlabeled data.

**Keywords:** Large scale image retrieval, annotation, transductive support vector machines, hashing, semi-supervised learning

## TEŞEKKÜR

Tez çalışması boyunca bilgi ve deneyimleri ile bana yol gösteren danışmanım Doç. Dr. Hakan ÇEVİKALP'e katkılarından dolayı teşekkür ederim.

Tezin laboratuvar çalışmaları aşamasında yardımcı olan çalışma arkadaşlarım Meltem YALÇIN, Merve ELMAS ve Araş. Gör. Şahin IŞIK'a yardımları için teşekkür ederim.

Her konuda sabırla yardımcı olan ve beni hep destekleyen annem Tezcan DÖNGEL, babam Ahmet Hakkı DÖNGEL ve kardeşim Tolga DÖNGEL'e desteklerinden dolayı çok teşekkür ederim.

Bu tez çalışması TÜBİTAK tarafından EEEAG-113E118 no'lu proje kapsamında desteklenmiştir.



## İÇİNDEKİLER

### Sayfa

<b>ÖZET</b> .....	VI
<b>SUMMARY</b> .....	VII
<b>TEŞEKKÜR</b> .....	VIII
<b>İÇİNDEKİLER</b> .....	IX
<b>ŞEKİLLER DİZİNİ</b> .....	XI
<b>ÇİZELGELER DİZİNİ</b> .....	XIV
<b>KISALTMALAR DİZİNİ</b> .....	XV
<b>1. GİRİŞ</b> .....	1
<b>2. LİTERATÜR ARAŞTIRMASI</b> .....	3
2.1. Metin Tabanlı İmge Erişimi .....	3
2.2. İçerik Tabanlı İmge Erişimi .....	5
2.3. Anlam Bilimsel Boşluk .....	8
<b>3. YEREL İMGE BETİMLEYİCİLERİ VE KODLANMASI</b> .....	9
3.1. Yerel İmge Betimleyicileri .....	9
3.1.1. Ölçekten bağımsız öznitelik dönüşümü (SIFT) .....	9
3.1.1.1. Ölçek uzayında uç değer (minimum-maksimum) tespiti .....	9
3.1.1.2. Kilit noktalarının konumlarının belirlenmesi .....	12
3.1.1.3. Yönelim tespiti .....	14
3.1.1.4. Kilit noktası betimleyicisi .....	15
3.1.2. Hızlandırılmış gürbüz öznitelikler (SURF) .....	16
3.1.2.1. Kilit nokta tespiti .....	16
3.1.2.2. Ölçek uzayı oluşturulması .....	17
3.1.2.3. İlgili noktası konumlandırılması .....	18
3.1.2.4. İlgili noktası betimleyicisi ve eşleme .....	18
3.1.3. Yerel ikili örüntü (LBP) .....	20
3.1.4. Yerel üçlü örüntü (LTP) .....	22
3.2. Yerel Özniteliklerin Kodlanması .....	23
3.2.1. Kelimeler çantası tabanlı imge betimleme .....	24
3.2.2. Fisher vektör ile imge betimleme .....	25

## İÇİNDEKİLER (devam)

	<u>Sayfa</u>
3.2.2.1. Fisher çekirdek yapısı.....	26
3.2.2.2. Fisher vektör ile imge gösterimi.....	27
3.2.3. Fisher vektör ile kelimeler çantası yönteminin karşılaştırması .....	28
<b>4. YEREL DUYARLI ÖZETLEME (LOCALITY SENSITIVE HASHING) .....</b>	<b>30</b>
4.1. Gösterim.....	32
4.2. Genel Yerel Duyarlı Özetleme.....	32
4.3. $l_p$ Normu için Yerel Duyarlı Özetleme .....	33
4.3.1. $p$ -kararlı dağılımları .....	33
4.3.2. Özet kümesi .....	34
4.4. Yerel Duyarlı Özetlemenin Parametreleri .....	35
4.5. Yerel Duyarlı Özetleme Algoritması .....	36
<b>5. YÖNTEM.....</b>	<b>38</b>
5.1. İmge Veri Tabanının Oluşturulması ve İmgelerin Betimlenmesi.....	38
5.1.1. Yarı-otomatik imge etiketleme aracı .....	38
5.1.2. Fisher vektör ile imgelerin betimlenmesi .....	42
5.2. Sınıf Hiyerarşilerinin Belirlenmesi .....	43
5.3. Transdaktif Eğitim .....	45
5.4. Özetleme Yöntemi .....	48
<b>6. BULGULAR VE TARTIŞMA .....</b>	<b>51</b>
6.1. Fisher Vektör ile İmgelerin Betimlenmesine Dair Deney .....	52
6.2. Sınıf Hiyerarşilerinin Belirlenmesine Dair Deneyler .....	54
6.3. Transdaktif Öğrenmeye Dair Deneyler.....	55
6.4. Yerel Duyarlı Özetleme Yöntemine Dair Deneyler.....	58
6.4.1. INRIA Holidays veri tabanı üzerindeki deneyler .....	60
6.4.2. Kentucky veri tabanı üzerindeki deneyler .....	62
6.4.3. UC Merced Land Use veri tabanı üzerindeki deneyler.....	65
6.4.4. CIFAR-10 veri tabanı üzerindeki deneyler.....	66
<b>7. SONUÇ VE ÖNERİLER.....</b>	<b>69</b>
<b>KAYNAKLAR DİZİNİ.....</b>	<b>72</b>

## ŞEKİLLER DİZİNİ

### Sekil

### Sayfa

2.1 Arama motoruna “mısır” yazıldığında, arama motoru hem sebze olan mısır resimlerini hem de ülke olan Mısır’a ait fotoğrafları döndürmektedir. Sadece sözcük bilgisi kullanılarak yapılan sorgular, çoğu zaman istenen imgelerin içeriğini tam olarak yansıtamamaktadır.....	4
2.2 Metin tabanlı imge erişimi sistemine ait gösterim.....	4
2.3 Yerel öznitelik tabanlı imge erişimi. Sol taraftaki sorgu imgesinden çıkarılan yerel öznitelikler veri tabanındaki imgelere ait yerel öznitelikler ile karşılaştırılarak sorgu imgesine en benzer imgeler geri döndürülmektedir. (a) En yakın uzaklık kullanılarak oluşturulan eşleşmeler. (b) Geometrik kısıtların uygulanması sonucu geriye kalan eşleşmeler .....	5
2.4 Yerel öznitelik tabanlı imge erişimi işlem ve saklama yükünün imge sayısı ile değişimi. ( Bu bilgi International Computer Vision Summer School (2011) kapsamında ders veren Josef Sivic’in sunumundan alınmıştır ve özellikle Facebook veri tabanındaki imge sayısı günümüz itibarıyla burada verilen sayıdan çok daha fazladır.).....	6
2.5 İçerik tabanlı imge erişimi sistemine ait gösterim.....	6
2.6 Anlam bilimsel boşluk probleminin bir örneği. (a) ve (c) resimleri benzer renk histogramına sahip olmalarına rağmen farklı anlamdadırlar. (a) ve (e) resimleri farklı renk histogramına sahip olmalarına rağmen benzer anlamdadırlar (Bannour, 2013). ...	8
3.1 Gauss piramidi.....	10
3.2 Gauss farkları akış şeması .....	11
3.3 Gauss fark matrisinin yerel minimum ve maksimumun noktalarının bulunmasına ilişkin örnek.....	12
3.4 Görüntü gradyanlarından kilit noktası betimleyicisinin oluşturulması. ....	15
3.5 (8,1), (4,2) ve (8,2) LBP operatörleri .....	20
3.6 LBP değerinin hesaplanmasının grafiksel ifadesi. Eşikleme işleminden sonra 8-bitlik LBP kodu oluşturulur. Daha sonra bu kodun onlu sistemdeki karşılığı hesaplanarak merkez pikselin LBP değeri bulunur.....	21
3.7 LTP değerinin hesaplanmasının grafiksel ifadesi.....	22
3.8 LTP kodunun pozitif ve negatif LBP koduna ayrılmasına ilişkin örnek.....	23

## ŞEKİLLER DİZİNİ (devam)

### Sekil

### Sayfa

- 3.9 Kelimeler çantası temelli imge betimlemenin şematik gösterimi. (a) Eğitim setindeki imgelerin öznitelikleri çıkarılarak öznitelik uzayı doldurulmuştur. Beyaz elipsler yerel öznitelik bölgelerini göstermekte, siyah noktalar ise öznitelik uzayında çıkarılan betimlenmiş noktaları belirtmektedir. (b) Çıkarılan öznitelikler görsel kelimeleri elde etmek için kümeleneştir. Görsel kelimeler küme merkezleridir ve büyük mavi dairelerle gösterilmiştir. Noktalı mavi çizgiler, seçilen küme merkezlerine dayalı Voronoi hücrelerini ifade etmektedir. (c) Verilen yeni imgenin her bir özniteliği için en yakın görsel kelime belirlenmiş ve her bir öznitelik belirlenen görsel kelimeye atanmıştır (şekilde  $w_i, i = 1, \dots, 4$  olarak gösterilen 4 tane görsel kelime vardır). (d) Son olarak imge, atamaları yansıtan histogram vektörü ile temsil edilmektedir (Grauman ve Leibe, 2009)..... 25
- 3.10 Fisher vektör ile imge betimleme. .... 28
- 3.11 (a) Kelimeler Çantası yönteminde  $k$ -ortalama kümeleme gösterimi. (b) Fisher vektör yönteminde gauss karışım modeli ile kümeleme gösterimi (Jakob, 2012). .... 29
- 4.1 Verinin rasgele izdüşümleri alınır. Her izdüşüm birkaç bit ile nicelenir (Fergus vd., 2008)..... 36
- 4.2 Yerel Duyarlı Özetleme algoritmasının gösterimi. Tüm veri tabanındaki imgeler özetlenerek özet tablosu oluşturulur. Aynı özet fonksiyonlarla sorgu imgesi de özetlenir. Benzer imgeler çarpışır ve çarpışan imgeler içinde yakın komşular aranır (Grauman ve Fergus, 2013)..... 37
- 5.1 Yarı-otomatik imge etiketleme aracının görünümü (Döngel ve Çevikalp, 2014)..... 40
- 5.2 (a) “Güncelle” düğmesi. (b) “Programı Başlat”, “Güncel Programı Başlat” ve “Resim Listesini Aç” sekmeleri. (c) Sorgu imgesinin gösterildiği alan. (Döngel ve Çevikalp, 2014)..... 41
- 5.3 (a) İmge isim listesinin gösterildiği alan. (b) “Etiket Yükle” düğmesi. (c) Sorgu imgesini etiketlemede kullanılabilecek etiketlerin listelendiği alan. (d) Sorgu imgesine uygun etiketlerin listelendiği alan (Döngel ve Çevikalp, 2014)..... 41
- 5.4 (a) Eklenebilecek diğer etiketlerin yazıldığı ve eklendiği alan. (b) Sorgu imgesine uygun etiketlerin kaydedilmesi için gerekli “Kaydet” düğmesi. (c) “Temizle” düğmesi. (Döngel ve Çevikalp, 2014) ..... 41
- 5.5 5 sınıfı ayıran ikili hiyerarşik karar ağacı. Karar ağacı 5’i terminal 4’ü iç düğüm olmak üzere toplam 9 düğüm içermektedir. Her bir iç düğümde sınıflar (1,2,3,6 nolu düğümler) kümeleme yöntemiyle iki gruba bölünmüştür. .... 44
- 5.6 Rastgele sınıflar ve örnekler seçilerek oluşturulan birden fazla karar ağacı örneği..... 45

## ŞEKİLLER DİZİNİ (devam)

<u>Sekil</u>	<u>Sayfa</u>
5.7 Transdaktif Destek Vektör Makineleri ile öğrenme. Sadece etiketli veri kullanıldığında destek vektör makineleri veriyi ayırmak için kesik çizgi ile gösterilen hiperdüzlemi döndürmektedir. Fakat etiketsiz veri (dairelerle gösterilen veriler) de işin içine katıldığında koyu renk ile belirtilen çok daha iyi bir ayırıcı hiperdüzlem bulunmaktadır.....	47
6.1 Duyarlılık ve geri çağırma gösterimi ( <a href="http://www.di.ens.fr/willow/events/cvml2011/materials/practical-classification/">http://www.di.ens.fr/willow/events/cvml2011/materials/practical-classification/</a> ) ....	52
6.2 PASCAL VOC 2007 veri tabanına ait çeşitli örnek imgeler.....	53
6.3 Yerel duyarlı özetleme yöntemine dair deneyde yapılan aşamalar .....	59
6.4 INRIA Holidays veri tabanına ait çeşitli örnek imgeler (Jegou ve Douze, 2008).....	60
6.5 Kentucky veri tabanına ait çeşitli örnek imgeler (Nister ve Stewenius, 2006). ....	63
6.6 UC Merced Land Use veri tabanına ait her bir sınıf için örnek imgeler .....	65
6.7 CIFAR-10 veri tabanına ait örnek imgeler .....	67

## ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
6.1 PASCAL VOC Veri tabanı üzerinde Fisher vektör betimlemesi kullanılarak elde edilen tanıma oranları.....	53
6.2 Caltech 256 veri tabanı üzerinde elde edilen tanıma oranları (%). .....	55
6.3 Caltech -256 üzerindeki deney sonuçları (Döngel ve Çevikalp, 2015).....	57
6.4 AR veri tabanı üzerindeki deney sonuçları (Döngel ve Çevikalp, 2015).....	57
6.5 INRIA Holidays veri tabanı üzerindeki deney sonuçları ( $L = 5, k = 50, w = 0$ ).....	61
6.6 INRIA Holidays +Flickr1M veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 15, w = 0$ ) .....	62
6.7 Kentucky veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 4$ ).....	64
6.8 Kentucky+Flickr1M veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 4$ ) .....	65
6.9 UC Merced Land Use veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 1$ )( $n = 250$ ).....	66
6.10 CIFAR-10 veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 0$ )( $n = 500$ ) .....	68

## KISALTMALAR DİZİNİ

### Kısaltmalar

### Açıklama

DoG	Difference of Gaussians
DVM	Destek Vektör Makineleri
E2LSH	Exact Euclidean Locality Sensitive Hashing
FT	Fourier Transform
FV	Fisher vektör
GKM	Gauss Karışım Modeli
LBP	Local binary patterns
LoG	Laplacian of Gaussian
LTP	Local Ternary Patterns
NCuts	Normalized Cuts
SIFT	Scale Invariant Feature Transform
SMO	Sequential Minimal Optimization
SURF	Speeded-Up Robust Features
TBA	Temel Bileşen Analizi
YDÖ	Yerel Duyarlı Özetleme

## 1. GİRİŞ

Bu tez kapsamında milyonlarca imgeden oluşan bir veri tabanı kullanarak etkin ve başarılı bir imge erişimi sistemi geliştirilmesi amaçlanmıştır. Literatürde bu amaçla önerilen birçok yöntem mevcuttur. Önerilen bu yöntemler arasında gerçekçi ve büyük veri tabanları kullanılarak geliştirilen yöntemler genellikle etiket bilgisini kullanmamaktadırlar. Bu yöntemler, uzaklık ölçütleri kullanarak giriş örnek uzayındaki uzaklıklara bakıp bu uzaklıkları yansıtan özet kodları üretme üzerine kuruludurlar. Ancak uzaklık ölçütü ne kadar iyi olursa olsun düşük seviye öznelilikler kullanılarak oluşturulan imge özneliliklerini kullandığından genellikle imgelerdeki anlam bilimsel ilişkileri yansıtmamaktadır (anlam bilimsel boşluk). Bu nedenle bu tür yöntemlerin başarı oranı oldukça düşüktür. Etiket bilgisini kullanan yöntemler ise ya eğitim kümesindeki sınırlı sayıda örneği kullanarak birçok verinin bilgisini kullanmazlar ya da gerçekçi veri tabanları yerine giriş uzay boyutu küçük ve gerçekçi olmayan veri tabanları kullanırlar. Bunun nedeni ise yöntemlerin sadece imgelere ait öznelilik vektörlerinin boyutu küçük olduğu durumlarda çalışabilmeleridir.

Bu tez çalışmasındaki amaç çeşitli yaklaşımlar kullanarak etiket bilgilerini en etkin bir şekilde kullanan imge erişim sistemi geliştirmek ve bu sistemi gerçekçi imge veri tabanları üzerinde sınamaktır. Bu amaç doğrultusunda öncelikle imgeler ve imgelerin etiket bilgisi kullanılarak sınıf hiyerarşileri oluşturulmuştur. Daha sonra transdaktif öğrenme ve büyük veri tabanlarına uygun yarı-eğitmenli toplama algoritmaları kullanılarak etiket bilgisi olmayan veriler etkin bir şekilde etiketlenmiştir. Ardından anlam bilimsel etiket bilgilerini yansıtan özet kodları oluşturulmuştur. Özet kodlar oluşturulduktan sonra, yeni bir sorgu imgesi geldiğinde ilk olarak özet kodu oluşturularak veri tabanındaki imgelerin özet kodları ile karşılaştırılmıştır. Bu sayede birkaç saniye gibi kısa bir zamanda veri tabanında bu sorgu imgesine benzeyen imgelerin döndürülmesi sağlanmıştır.

Bu tez çalışması 7 temel bölümden oluşmaktadır. İlk bölümde çalışmanın amacı ve kapsamından bahsedilmiştir. İkinci bölümde imge erişimi için kullanılan yöntemlerden ve karşılaşılan zorluklardan kısaca bahsedilmiştir. Üçüncü bölümde imge betimlemek için



kullanılan yerel öznitelikler ve bunların kodlanması hakkında bilgiler verilmiştir. Dördüncü bölümde büyük veri tabanlı imge erişimi için kullanılan özetleme algoritmalarının detayları anlatılmıştır. Beşinci bölümde büyük veri tabanlı imge erişimi için önerdiğimiz sistem açıklanmış olup, deneysel çalışmalar altıncı bölümde anlatılmıştır. Son bölümde ise vargılar verilerek tez sonlandırılmıştır.

## 2. LİTERATÜR ARAŞTIRMASI

Son yıllarda hızla gelişen dijital fotoğraf çekme teknolojisi sonucunda akıllı telefon, tablet gibi birçok elektronik cihazda fotoğraf çekme standart bir hal almıştır. Buna paralel olarak kolayca taşınabilen dijital kameraların ve cep telefonlarının fotoğraf çekme amacıyla kullanılması ile insanlar hemen hemen her yerde ve her zaman fotoğraf çekebilmektedir. Sosyal ağların fotoğraf ve video paylaşımına izin vermesi sonucunda da insanlar çektikleri fotoğraf ve videoları internet ortamında hızlıca paylaşabilmektedir. Bu yüzden internet ortamındaki görsel imgelerin sayısında çok büyük bir artış yaşanmıştır. Örnek vermek gerekirse Facebook ortamındaki resim sayısı 2010 yılında  $10^{10}$  civarında olup bu sayı gün geçtikçe hızla artmaktadır. Buna Google, Flickr gibi sitelerde eklendiğinde internet ortamındaki fotoğraf sayısı trilyonlarla ifade edilebilmektedir. İnternet ortamındaki fotoğraf sayısının hızla artmasının aksine büyük imge koleksiyonlarını yönetmek için gerekli yazılım oldukça sınırlıdır. İmge koleksiyonlarının yönetimi ile ilgili en büyük zorluk muhtemelen imge erişimi konusudur. Bu konuyla ilgili olarak birçok girişimde bulunulmuştur. Mevcut olan yöntemler iki kategoriye ayrılabilir: metin tabanlı imge erişimi ve içerik tabanlı imge erişimidir. İlerleyen bölümlerde bu tekniklerin avantajları, dezavantajları ve eksiklikleri tartışılacaktır.

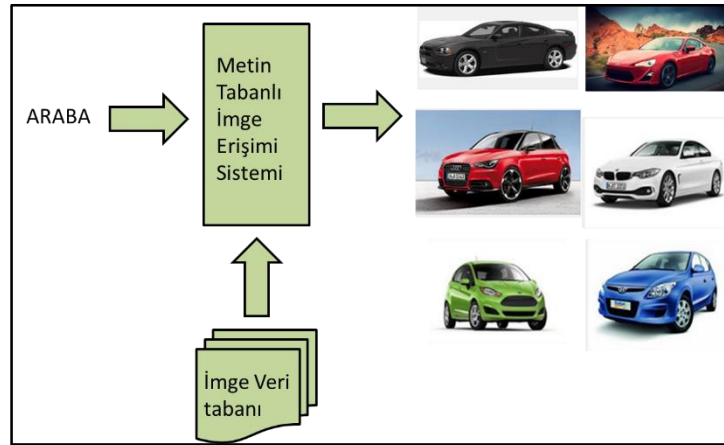
### 2.1. Metin Tabanlı İmge Erişimi

Yakın bir zamana kadar imge erişimi genellikle metin tabanlı olarak yapılmıştır. Metin tabanlı imge erişimi, manuel olarak etiketlenen imgelere dayanmaktadır. Kullanıcılar her bir imgeyi manuel olarak, imge içeriğini bir veya birden fazla anahtar kelimeyle açıklayacak şekilde etiketleme işlemini yapmaktadırlar. Manuel etiketleme işlemi hem zaman alıcı hem de öznel bir iştir ve kişiden kişiye farklılık gösterebileceği gibi aynı kişi farklı yer ve zamanlarda aynı imgeyi farklı ifadeler kullanarak etiketleyebilmektedir. Bu da imge erişiminde bazı sorunlar doğurmaktadır. İnternet ortamındaki tüm imgelerin doğru bir şekilde insanlar tarafından manuel olarak etiketlenmesi mümkün değildir. Bu tür etiketleme işlemi yarı-otomatik veya tam-otomatik algoritmalar tarafından yapılmakta ve genellikle etiketler imgeyle beraber gelen metinden seçilen anahtar kelimelerden türetilmektedir. Bu sebeple bu etiketler gürültü de

içermektedir. Bunun yanında sadece sözcük bilgisi çoğu zaman bulunmak istenen imgelerin içeriğini tam olarak yansıtamamaktadır. Örneğin Şekil 2.1’de gösterilen arama motoruna “mısır” yazıldığında, arama motoru hem mısır resimleri hem de Mısır’a ait fotoğrafları döndürmektedir. Bu tür sorunların önüne geçmek için sözcükle birlikte sorgu resimleri ya da sadece sorgu resimleri kullanılmaya başlanmıştır. Şekil 2.2’de metin tabanlı imge erişimi sistemine ait bir örnek gösterilmektedir.



Şekil 2.1 Arama motoruna “mısır” yazıldığında, arama motoru hem sebze olan mısır resimlerini hem de ülke olan Mısır’a ait fotoğrafları döndürmektedir. Sadece sözcük bilgisi kullanılarak yapılan sorgular, çoğu zaman istenen imgelerin içeriğini tam olarak yansıtamamaktadır.



Şekil 2.2 Metin tabanlı imge erişimi sistemine ait gösterim

## 2.2. İçerik Tabanlı İmge Erişimi

İmge sayısının az olduğu durumlarda imge erişimi için kullanılan en başarılı yöntemler yerel öznitelikleri kullanmaktadır (Schmid ve Mohr, 1997; Sivic ve Zisserman, 2003; Mikolajczyk ve Schmid 2002; Lowe, 1999 ). Bu düşük seviye öznitelik vektörleri renk, şekil, doku gibi temel imge bilgilerini içermektedir. Bu tür yaklaşımlarda ilk olarak Harris, Laplace, DoG (Difference of Gaussians) gibi belirgin nokta bulma algoritmaları (Lowe, 2004; Mikolajczyk ve Schmid, 2004; Mikolajczyk ve Schmid, 2005) kullanılarak imgelerdeki önemli noktalar belirlenmiştir. Daha sonrasında bu bölgelerin etrafından seçilen yamalar SIFT betimleyicileriyle 128 boyutlu sabit vektörel değerlere dönüştürülmüştür. Eğitim kümesindeki her bir imge bu çok sayıdaki yama vektörü ile ifade edilmiştir. Bir sorgu imgesi geldiğinde, aynı algoritmalar bu imge üzerinde de çalıştırılarak yama vektörleri elde edilmiş ve daha sonra bu yama vektörleri eğitim kümesindeki yama vektörleri ile karşılaştırılarak en yakın yamalar bulunmuştur. Bu işlemden sonra yerel ve global geometrik kısıtlar kullanılarak elemeler yapılmış ve sorgu imgesine en benzer imgeler belirlenmiştir. Bu işlem Şekil 2.3’de gösterilmiştir.



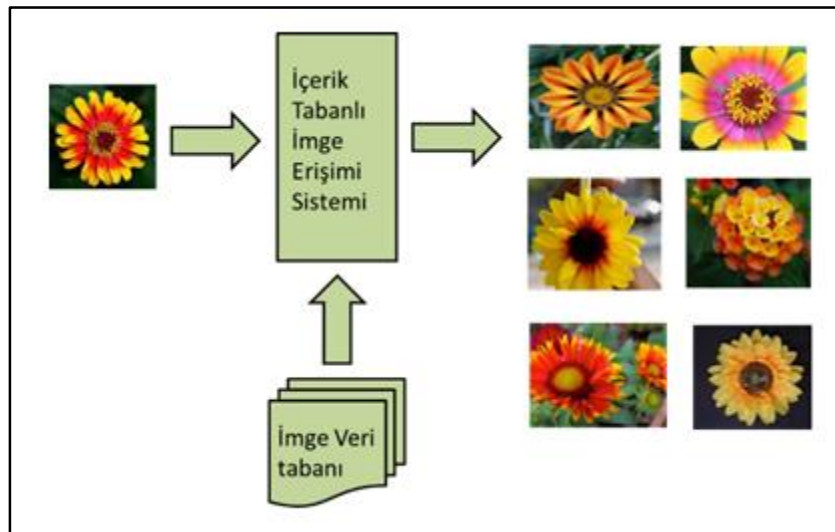
Şekil 2.3 Yerel öznitelik tabanlı imge erişimi. Sol taraftaki sorgu imgesinden çıkarılan yerel öznitelikler veri tabanındaki imgelere ait yerel öznitelikler ile karşılaştırılarak sorgu imgesine en benzer imgeler geri döndürülmektedir. (a) En yakın uzaklık kullanılarak oluşturulan eşleşmeler. (b) Geometrik kısıtların uygulanması sonucu geriye kalan eşleşmeler

Bu yaklaşımı veri tabanındaki imge sayısının çok fazla olduğu durumlarda kullanmak hem gerçek zamanlı işlem yükü hem de hafıza kısıtları yüzünden mümkün olmadığı Şekil 2.4’de görülmektedir. Örnek olarak eğitim kümesinde 10 milyar imgemiz olduğunu varsayarsak ve her imgeden ortalama 1000 civarı yama seçilirse 128 boyutlu

yama vektörlerinin sayısı 10 trilyon civarında olacaktır. Bu vektörleri hard diske sığdırmak mümkün olmadığı gibi bir sorgu imgesine benzer imgelerin döndürülmesi ortalama bir donanımla 300 yıl alacaktır. Şekil 2.5’de içerik tabanlı imge erişimi sistemine ait bir örnek gösterilmektedir.

<b>N ... images</b>		
<b>M ... regions per image (~1000) (yama sayısı)</b>		
<b>D ... dimensions of the descriptor (~128)</b>		
<b>Exhaustive linear search: <math>O(M \cdot N \cdot D)</math></b>		
<b>Example:</b>		
• <b>Matching two images (N=1) each having 1000 SIFT descriptors</b>		
• <b>Nearest neighbours search: 0.4 s (2 GHz CPU, implementation in C)</b>		
• <b>Memory footprint: <math>1000 * 128 = 128 \text{ kB} / \text{image}</math></b>		
<b># of images</b>	<b>CPU time</b>	<b>Memory req.</b>
N = 1,000 ...	~7 min	(~100 MB)
N = 10,000 ...	~1 h 7 min	(~ 1 GB)
...		
N = $10^7$	~115 days	(~ 1 TB)
...		
<b>All images on Facebook :</b>		
N = $10^{10}$ ...	~300 years	(~100 MB)

Şekil 2.4 Yerel öznitelik tabanlı imge erişimi işlem ve saklama yükünün imge sayısı ile değişimi. ( Bu bilgi International Computer Vision Summer School (2011) kapsamında ders veren Josef Sivic’in sunumundan alınmıştır ve özellikle Facebook veri tabanındaki imge sayısı günümüz itibarıyla burada verilen sayıdan çok daha fazladır.)



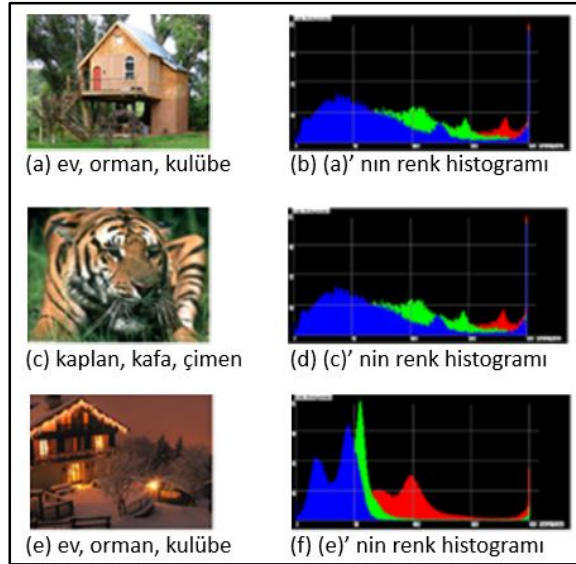
Şekil 2.5 İçerik tabanlı imge erişimi sistemine ait gösterim

Yukarıda bahsedilen problemi çözmek için her bir imgeden elde edilen yama vektörlerine nicemleme işlemi uygulanır. Kelimeler çantası (Bag of Words - BoW) tabanlı betimleme (Csurka, 2004; Jurie ve Triggs, 2005; Fergus, 2005; Harzallah, 2009) olarak da bilinen bu işlem sırasında tüm imgelerden elde edilen milyarlarca yama vektörüne  $k$ -en yakın merkez gibi bir toplama algoritması uygulanır ve her bir imgeden gelen yama vektörleri  $k$  sayıdaki topak merkezlerinden birine atanır. Atama işlemi tamamlandıktan sonra her bir topak merkezine gelen oylar sayılarak toplam oy sayısına bölünerek normalizasyon uygulanır ve sayısal imge bu atamaları gösteren histogram ile ifade edilir. Bu şekilde her bir imge  $k$ -boyutlu bir öznitelik vektörü (histogram vektörü) ile ifade edilmiş olur. Bu durumda imgelerin boyutu (MD)'den  $k$ -boyuta düşürülmüş olur. İmge sayısının çok fazla olduğu uygulamalarda başarılı sonuçlar için  $k$ 'nın değeri genellikle 100K ve katları olacak şekilde seçilmiştir. Birçok araştırmacı yukarıda ana hatlarıyla verilen yaklaşımı izleyerek, nicemleme hatalarını azaltmak için iyileştirmeler yaparak ya da yamalar arasındaki zayıf geometrik ilişkileri kullanarak bu yaklaşımın değişik türevlerini önermişlerdir.

Kelimeler çantası tabanlı imge betimleme hem işlem hem de saklama yükünü büyük ölçüde azaltsa da görsel kelimelere karşılık gelen topak merkezlerinin sayısının ( $k$ ) fazla olduğu durumlarda veri tabanındaki imge sayısı da fazla ise bu sorunları tam olarak çözemez. İmge histogramlarını daha etkili bir şekilde hafızada saklamak için ters çevrilmiş dosya indisi (inverted file index) gibi bu histogramları daha az yer kaplayacak şekilde saklayan formatlara başvurulur. Fakat daha iyi bir çözüm imge vektörlerini ikili (binary) Hamming uzayına aktarmaktır. Bu şekilde her bir imge  $b$ -bitten oluşan ve özet (hash) vektörü adı verilen bir vektör ile ifade edilir (bu işlem özetleme olarak da adlandırılır). Özet vektörleri hafızada daha az yer tutacak bir şekilde saklanır. Ayrıca bit düzeyinde işlemler CPU tarafından çok hızlı bir şekilde gerçekleştirildiğinden, Özetleme (hashing) tabanlı imge betimleme sorgu imgesine benzer imgeleri arama işini oldukça hızlandırır. Özetleme işleminde amaç  $k$  boyutlu örnek uzayında imge histogramlarının arasındaki uzaklıkları koruyan özet kodları üretmektir (bu sebeple yöntem genellikle Yerel Duyarlı Özetleme-Locality-Sensitive Hashing olarak adlandırılır).

### 2.3. Anlam Bilimsel Boşluk

İmge erişimi için önerilen ilk yöntemlerde imgelerin düşük seviye öznitelik vektörlerinin çıkarılması ve imge öznitelikleri arasındaki benzerliğin ölçülmesine dayalı teknikler kullanılmıştır. Ancak bu yöntemlerle imgeleri doğru bir şekilde sınıflandırmak ya da imgenin içeriğini doğru bir şekilde anlamak pek mümkün değildir. Bu sorun imge erişiminin en büyük zorluğu olarak bilinen anlam bilimsel boşluk problemidir. Şekil 2.6'da gösterildiği üzere farklı resimler farklı anlamlara sahip olmalarına rağmen benzer histogramlara sahip olabilecekleri gibi benzer resimler de aynı anlama sahip olmalarına rağmen farklı histogramlara sahip olabilirler. Bu da imgeleri doğru bir şekilde sınıflandırma yapılamamasına yol açmaktadır.



Şekil 2.6 Anlam bilimsel boşluk probleminin bir örneği. (a) ve (c) resimleri benzer renk histogramına sahip olmalarına rağmen farklı anlamdadırlar. (a) ve (e) resimleri farklı renk histogramına sahip olmalarına rağmen benzer anlamdadırlar (Bannour, 2013).

### 3. YEREL İMGE BETİMLEYİCİLERİ VE KODLANMASI

Nesne tanıma ve imge erişimi genel olarak üç temel adımdan oluşmaktadır : (i) yerel imge özniteliklerinin çıkarılması (SIFT, SURF, LBP, LTP betimleyicileri gibi), (ii) bir imge betimleyicisinde yerel özniteliklerinin kodlanması (örneğin nicemlenmiş yerel özniteliklerinin histogramı) ve (iii) imge betimleyicilerinin sınıflandırılması (DVM gibi). Bu bölümde ilk olarak yerel imge betimleyicilerden bahsedilmiş daha sonrada bunların kodlanması hakkında bilgi verilmiştir.

#### 3.1. Yerel İmge Betimleyicileri

İmge erişimi için gerekli olan üç temel adımdan ilki yerel imge özniteliklerinin çıkarılmasıdır. Literatürde bu amaç için birçok yöntem önerilmiştir. Bunlardan bazıları SIFT, SURF, LBP, LTP betimleyicileri gibi yöntemlerdir. Bu bölümde yerel imge betimleyicileri detaylı bir şekilde anlatılacaktır.

##### 3.1.1. Ölçekten bağımsız öznitelik dönüşümü (SIFT )

SIFT (Scale Invariant Feature Transform), David Lowe (2004) tarafından geliştirilmiş etkili bir ilgi noktası tespit yöntemidir. SIFT algoritmasıyla oluşturulan betimleyici vektörler ölçekten ve rotasyondan bağımsızdır. SIFT algoritması aşağıda verilen 4 temel adımdan oluşmaktadır (Lowe, 2004):

1. Ölçek uzayında uç değer (minimum-maksimum) tespiti
2. Kilit noktalarının konumlarının belirlenmesi
3. Yönelim tespiti
4. Kilit noktası betimleyicisi

##### 3.1.1.1. Ölçek uzayında uç değer (minimum-maksimum) tespiti

Uç değer, ölçeksel uzayda sabit, değişmeyen noktalardır. Ölçeksel uzay oluşturarak kilit nokta olabilecek noktaları bulmak için birçok teknik kullanılabilir. Bunlardan birisi de DoG (Difference of Gaussians) yöntemidir. Koenderink (1984) ve Lindeberg



(1994); yaptıkları çeşitli varsayımlar sonucunda ölçeksel uzayın Gauss fonksiyonu kullanılarak oluşturulabileceğini tespit etmişlerdir. SIFT algoritmasında da, DoG metodu kullanılmıştır.

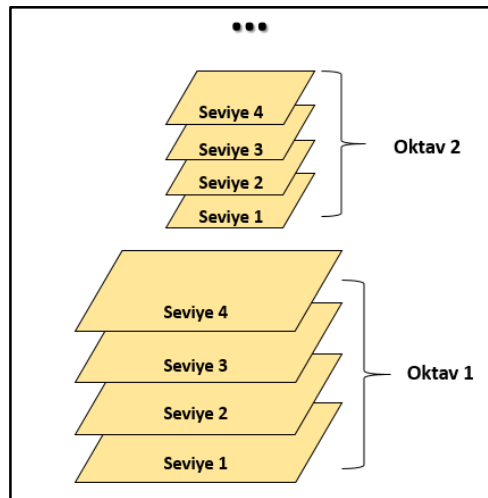
Ölçeksel uzayda uç değer tespiti kapsamında ilk olarak ölçeksel uzay oluşturulmaktadır. Bunun için ilk başta orjinal görüntü  $I(x, y)$ , farklı standart sapmaya ( $\sigma$ ) sahip Gauss filtrelerden geçirilerek Gauss piramidi oluşturulmaktadır. Görüntünün filtrelenmesinde kullanılan Gauss filtre aşağıdaki denklemde gösterilmiştir:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3.1)$$

$I(x, y)$  görüntüsünün Gauss filtresinden geçirilmesiyle elde edilen filtrelenmiş görüntü denklem 3.2'deki gibidir:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.2)$$

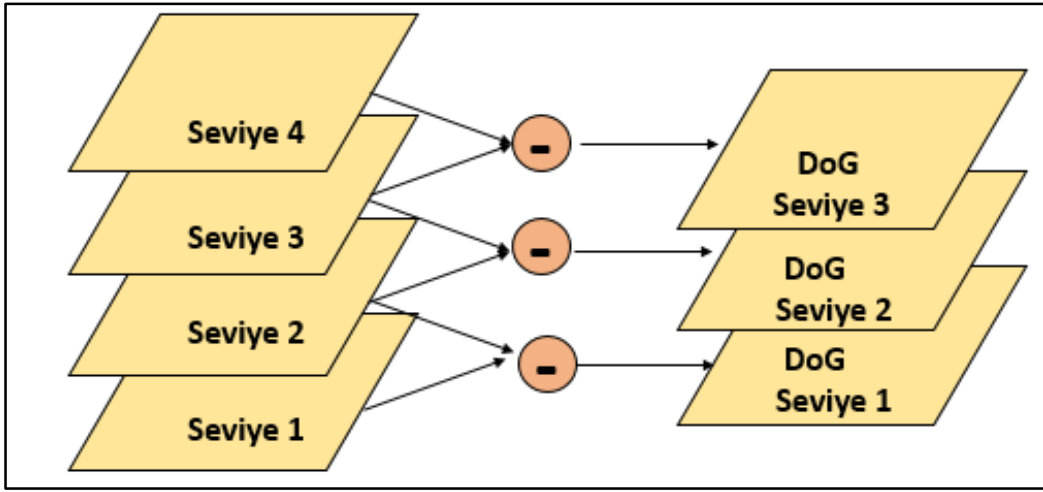
$L(x, y, \sigma)$  değeri  $\sigma$  değeri için Gauss filtre çıktısını belirtmektedir. Bu işlem farklı  $\sigma$  değerleri için uygulanır ve bir Gauss piramidinin bir oktav için katmanı elde edilir. Görüntü boyutu birkaç kez yarıya indirilerek farklı oktav değerleri için Gauss filtreleme işlemi yapılmaktadır. Bu işlemler Şekil 3.1'deki akış şemasında gösterilmiştir.



Şekil 3.1 Gauss piramidi

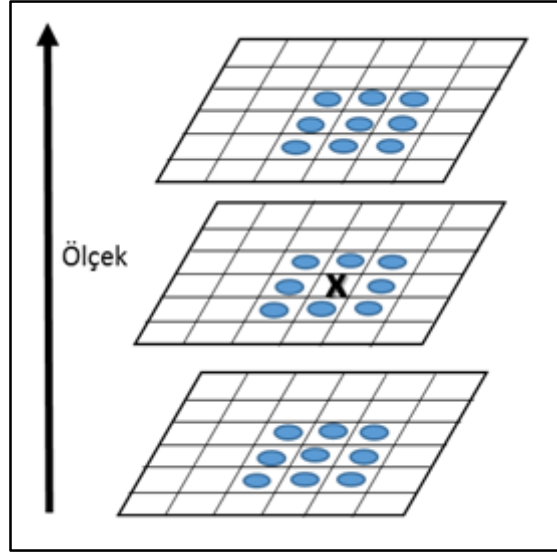
Görüntü farklı oktavlarda ve farklı  $\sigma$  değerlerinde filtrelenerek Gauss piramidi oluşturulmuştur. Gauss piramidi oluşturulduktan sonra her bir oktav ve  $\sigma$  seviyesi için DoG matrisleri elde edilir. Bunun için her bir oktavda  $\sigma$  seviyeleri arasındaki fark bulunur. Bu işlem uygulandıktan sonra her bir fark işlemi için bir adet  $D(x, y, \sigma)$  fark matrisi denklem 3.3'deki gibi elde edilir. Bu işlemler Şekil 3.2'deki akış şemasında gösterilmiştir.

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (3.3)$$



Şekil 3.2 Gauss farkları akış şeması

$D(x, y, \sigma)$  fark matrisinin yerel minimum ve maksimumunu tespit edebilmek için, bir görüntüde her bir piksel alt ve üst ölçeğinde 9'ar komşusu ve ana ölçekte 8 komşusu olmak üzere toplam 26 komşusu ile karşılaştırılmaktadır. Tüm komşulardan büyük veya küçük piksel değerine sahip olan noktalar bulunur. Bu durum Şekil 3.3'de gösterilmiştir:



Şekil 3.3 Gauss fark matrisinin yerel minimum ve maksimum noktalarının bulunmasına ilişkin örnek.

### 3.1.1.2. Kilit noktalarının konumlarının belirlenmesi

Bir önceki adımda, bir pikseli komşularıyla karşılaştırarak birçok kilit nokta adayı bulunmuştur. Bu kilit noktalardan bazıları kararlı değildirler yani düşük kontrasta sahiptirler ve dolayısıyla gürültüden daha çok etkilenirler, ya da kenarlardadırlar. Her iki durumda da, bunlar öznitelik olarak kullanışlı değildir. Bu yüzden bu kilit noktalarının ayıklanması gereklidir.

Düşük kontrasta sahip noktaların ayıklanması, DoG fonksiyonunun ( $D(x, y, \sigma)$ ) kilit aday nokta baz olmak üzere ikinci dereceden Taylor serisi açılımı yapılarak gerçekleştirilir ve aşağıdaki eşitlik elde edilir:

$$D(x) = D + \frac{\partial D}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (3.4)$$

burada DoG fonksiyonu  $D(x)$  ve onun türevi, her uç nokta için hesaplanır. Uç noktalarının yeni konumları,  $\hat{x}$

$$\hat{x} = - \frac{\partial^2 D^{-1} \partial D}{\partial x^2 \partial x} \quad (3.5)$$

ile hesaplanır ve

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (3.6)$$

elde edilir.  $|D(\hat{x})| < 0.03$  ise bu nokta elenir.

Kararlılık için sadece düşük kontrastları reddetmek yeterli değildir. Gauss fonksiyonu farkı kenar noktalarda yüksek tepkiler vermektedir. Bu sebeple kilit nokta adayları kenarlarda ise bu noktalar da elenmelidir.

Gauss fonksiyonu farkında yetersiz tanımlanmış tepe noktaları, kenar boyunca büyük bir ana eğriliği, ancak dikey yönde küçük bir ana eğriliği olacaktır. Temel eğriler, kilit nokta  $2 \times 2$  lik Hessian matrisi,  $\mathbf{H}$ , ile hesaplanır.

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (3.7)$$

Türevleri, komşu örnek noktaların farkları alınarak tahmin edilmektedir. Hessian matrisinin öz değerleri,  $D$ 'nin ana eğriliği ile orantılıdır.  $\alpha$  büyük olan özdeğer ve  $\beta$  da küçük olan özdeğer olarak seçilirse  $r$  de bunlar arasındaki orandır ( $\alpha = r\beta$ ).

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta, \quad (3.8)$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (3.9)$$

Ana eğrilik oranı, belli bir eşik değerinin,  $r$ , altında olup olmadığını kontrol etmek için aşağıdaki denklem kullanılır:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (3.10)$$

Eşik değeri  $r = 10$  kullanılarak kenarlarda tespit edilen kilit noktalar elenir.

### 3.1.1.3. Yönelim tespiti

Algoritmanın bu adımında her bir kilit noktaya, bölgesel özelliklerine dayalı uygun bir yönelim belirlenip, kilit noktası yönelimliyle birlikte tanımlanmıştır. Böylece görüntünün dönmesine karşı değişimsizlik kazanılır. Kilit noktanın yönelimini atamak için kullanılan yöntemlerden en kararlısı bir bölge seçilip bu bölgenin ağırlıklandırılarak yöneliminin tanımlanması olarak bulunmuştur. Bunun için anahtar noktasına en yakın ölçekteki görüntü seçilip bütün ölçek bağımsız hale getirilmektedir.

Gauss filtresinden geçirilmiş her bir resimde bulunan kilit noktaların bu ölçekte, komşu her pikseli için gradyen büyüklüğü,  $m(x, y)$ , ve yönelimi,  $\theta(x, y)$ , piksel farkları kullanılarak hesaplanır:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (3.11)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}\right) \quad (3.12)$$

Kilit noktaların etrafında her biri 10 derecelik bir yönü kapsayan 36'lık bir yönelim histogramı oluşturulur. Histograma eklenen her örnek nokta, kendi gradyen büyüklüğü ve kilit nokta ölçeğinde  $\sigma$ 'nın 1,5 katına eşit olan Gauss ağırlıklı dairesel pencere ile ağırlıklandırılır.

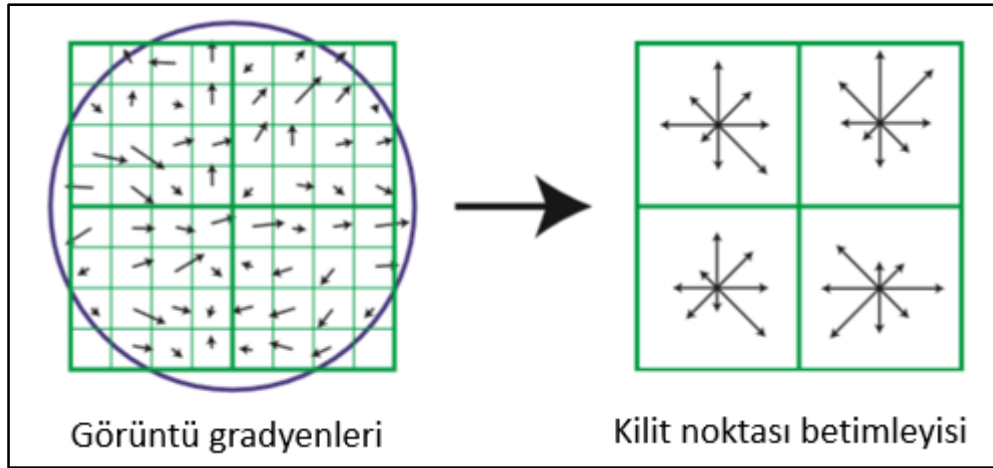
Yönelim histogramında tepe noktaları, yerel gradyenlerinin baskın yönlerine karşılık gelmektedir. Histogramdaki en büyük değere sahip nokta tespit edilir ve ardından %80'i içinde kalan diğer yerel maksimum noktaları da, yönelim ile kilit nokta oluşturmak için kullanılır. Böylece benzer büyüklükte birden fazla tepe noktalı yerler için, aynı yerde ve ölçekte ama farklı yönelime sahip birden fazla kilit nokta olacaktır.

Noktaların sadece yaklaşık %15'i için çoklu döngüsel atama yapılmış olmasına rağmen bu işlem eşleşmelerdeki kararlılığı önemli ölçüde arttırmaktadır.

### 3.1.1.4. Kilit noktası betimleyicisi

Bundan önceki adımlarda her kilit noktasına bir görüntü konumu, ölçeği ve yönelimi atanmıştır. Bu adım da ise parlaklık değişimleri, üç boyutlu görünüm gibi değişimlere karşı dayanıklılık kazandırılmaktadır.

İlk olarak kilit nokta etrafındaki  $16 \times 16$ lık komşuluk alınır. Bu  $4 \times 4$  boyutunda 16 alt blok halinde ayrılır. Her bir alt blok için, 8 aralık(bin) yönelim histogramı örneklerin büyüklük ve yönelim değerlerinden hesaplanarak oluşturulur. Büyüklükleri, betimleyici pencere genişliğinin 1,5 katına eşit Gauss fonksiyonu  $\sigma$  ile ağırlıklandırılır. Betimleyici daha sonra bu histogramların bütün değerlerinden oluşan bir vektör haline getirilir.  $4 \times 4 = 16$  tane histogram ve her birinde 8 tane aralık olduğundan her kilit noktası için öznitelik vektöründe  $4 \times 4 \times 8 = 128$  öge vardır. Dolayısıyla her bir kilit nokta için 128 boyutlu öznitelik vektörü tanımlanmaktadır. Görüntü gradyanlarından kilit noktası betimleyicisinin oluşturulması Şekil 3.4'te gösterilmiştir.



Şekil 3.4 Görüntü gradyanlarından kilit noktası betimleyicisinin oluşturulması.

Son olarak öznitelik vektörü, aydınlatma değişikliklerinin etkisini azaltmak için birim uzunluğa normalleştirilir. Doğrusal olmayan aydınlatma etkisini azaltmak için 0,2 eşik değeri uygulanır ve vektör tekrar normalleştirilir.

### 3.1.2. Hızlandırılmış gürbüz öznitelikler (SURF)

SURF (Speeded-Up Robust Features) Hızlandırılmış Gürbüz Öznitelikler ölçek ve yönelim değişimlerinden etkilenmeyen, nesne tanıma gibi görevler için kullanılacak bir yerel öznitelik betimleyicisidir. SURF yöntemi ilk olarak 2006 yılında Herbert Bay vd. (2006) tarafından ortaya atılmıştır. Kısmen SIFT tanımlayıcısından esinlenilmiştir. SURF'ün standart versiyonu çoğu kez SIFT'den daha hızlıdır ve farklı görüntü dönüşümlerine karşı SIFT'e göre daha sağlam olduğu kendi yazarları tarafından iddia edilmektedir. SURF, 2 boyutlu Haar dalgacık tepkilerinin toplamına dayanmaktadır.

#### 3.1.2.1. Kilit nokta tespiti

Bu algoritmada kilit nokta tespiti için çok temel Hessian matris yaklaşımı kullanılmaktadır. Hesaplama süresini azaltmak için tamamlayıcı (integral) görüntüler kullanılmaktadır. Tamamlayıcı görüntüler verilen dikdörtgen bir alanın hızlı bir şekilde hesaplanmasını sağlamaktadır.  $\mathbf{x} = (x, y)^T$  konumunda tamamlayıcı görüntü  $I_{\Sigma}(\mathbf{x})$ , nokta,  $\mathbf{x}$ , ve onun merkezinde şekillenmiş dikdörtgen bir alan içindeki giriş görüntüsündeki,  $I$ , bütün piksellerin toplamı olarak ifade edilir.

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (3.13)$$

Tamamlayıcı görüntüler hesaplanır hesaplanmaz, herhangi bir dik dikdörtgen alanının üzerindeki yoğunlukların toplanması için 3 işlem yapılmaktadır. Dolayısıyla, hesaplama zamanı boyuttan bağımsız bir hal almaktadır.

SURF detektörü Hessian matrisinin determinantına dayanmaktadır. Hessian matrisinin determinantının maksimum olduğu konum görüntü bölgelerinin (blob) elde edilmesinde kullanılmaktadır. Ölçek ve konum seçimi için iki farklı yöntem kullanılmaktan ziyade, her ikisi içinde Hessian matrisinin determinantı kullanılmaktadır.  $I$  görüntüsünde  $\mathbf{x} = (x, y)$  noktası için Hessian matrisi  $H(\mathbf{x}, \sigma)$  aşağıdaki gibi tanımlanmıştır:

$$H(\mathbf{x}, \boldsymbol{\sigma}) = \begin{bmatrix} L_{xx}(\mathbf{x}, \boldsymbol{\sigma}) & L_{xy}(\mathbf{x}, \boldsymbol{\sigma}) \\ L_{xy}(\mathbf{x}, \boldsymbol{\sigma}) & L_{yy}(\mathbf{x}, \boldsymbol{\sigma}) \end{bmatrix} \quad (3.14)$$

Denklem 3.14'de verilen  $L_{xx}(\mathbf{x}, \boldsymbol{\sigma})$ , Gauss fonksiyonunun 2. Dereceden türevi  $\frac{\partial^2}{\partial x^2} g(\sigma)$  ile  $I$  görüntüsünün  $\mathbf{x}$  noktasında konvolüsyonudur ayrıca  $L_{xy}(\mathbf{x}, \boldsymbol{\sigma})$  ve  $L_{yy}(\mathbf{x}, \boldsymbol{\sigma})$  değerleri de benzer şekilde hesaplanmaktadır.

Tahmini Gausslar kullanılarak Hessian matrisinin determinanı denklem 3.15 ile hesaplanmaktadır. LoG (Laplacian of Gaussian) yaklaşımı ile bulunan 2.dereceden Gauss fonksiyonunun kısmi türevleri  $x, y, xy$  yönünde ağırlıklandırılmış kutu filtre yaklaşımı  $D_{xx}, D_{xy}, D_{yy}$  olarak gösterilmektedir.

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2. \quad (3.15)$$

Filtre tepkilerinin bağıl ağırlığı,  $w$ , Gausslar arasındaki enerji dönüşümü için gereklidir ve bu değer Bay tarafından 0,9 olarak önerilmektedir.

### **3.1.2.2. Ölçek uzayı oluşturulması**

Eşleştirilmesi istenen görüntüler farklı ölçeklerde olabilmekte ya da detaylar farklı ölçeklerden elde edilebilmektedir. Ölçek uzayı genellikle görüntü piramitleri şeklindedir. Görüntüler defalarca bir Gaussian ile yumuşatılmakta, sonra piramidin en üst seviyesine ulaşmak için alt örneklenmektedir. Lowe (2004) kenarları ve imge bölgeleri bulunabilen görüntünün Gauss farkını almak için bu piramit katmanlarını çıkarmaktadır.

Kutu filtreler ve tamamlayıcı görüntüler kullanılması sebebiyle, önceden filtrelenmiş katmanın çıkış verisine aynı filtre iteratif olarak uygulanmamaktadır. Ancak bunun yerine doğrudan orijinal görüntü üzerinde tam olarak aynı hızda herhangi bir boyutta kutu filtre uygulanabilmektedir. Bu sebeple görüntü boyutunu iteratif olarak azaltmak yerine ölçek uzayı üst ölçeklendirme ile analiz edilmektedir.



Ölçek uzayı oktavlara bölünürken bir oktav artan boyutta bir filtre ile aynı giriş görüntüsünün konvolüsyonuyla elde edilen filtre tepki haritalarının bir dizisi olarak temsil edilmektedir. Toplamda, bir oktav 2'lik bir ölçeklendirme faktörünü kapsamaktadır ve her oktav ölçek seviyeleri sabit sayıda olacak şekilde bölünmüştür.

### **3.1.2.3. İlgili noktası konumlandırılması**

Görüntüdeki ve ölçeklerdeki ilgili noktaların konumlandırılması için  $3 \times 3 \times 3$  komşuluğunda maksimum olmayan noktaların elenmesi yöntemi uygulanmaktadır. Bu şekilde ilgili noktası adayları elde edilmektedir. Ölçek uzayında bu aşamanın gerçekleştirilmesi için her piksel değeri alt ve üst ölçeğinde bulunan 9'ar komşusu ve ana ölçekte bulunan 8 komşusu olmak üzere toplam 26 komşusu ile karşılaştırılmakta ve maksimum olmayan noktalar elenmektedir. Konumlandırmadaki son aşamada Hessian matrisinin determinantının maksimumları görüntü uzayında ve ölçekte noktası konumlarını bulmak için Brown ve Lowe (2002)'un önerdiği yöntem ile ara değerlendirilmektedir.

### **3.1.2.4. İlgili noktası betimleyicisi ve eşleme**

SURF betimleyicisi belirlenen her bir ilgili noktasının bir ölçeğe bağlı komşuluğu içerisinde piksel yoğunluklarının nasıl dağıtıldığını tanımlamaktadır. Bu yaklaşım SIFT'e benzemektedir. Fakat Haar dalgacık olarak bilinen ve  $x$  ve  $y$  gradyanlarını bulmak için kullanılabilen filtreler ile birlikte kullanılan tamamlayıcı görüntüler hesaplama süresini azaltmakta ve sağlamlığı arttırmada kullanılmaktadır.

İlk olarak, ilgili noktası etrafındaki dairesel bir bölgeden gelen bilgilere dayanarak görüntü yönelimine karşı değişmezliğini sağlamak amacıyla tekrarlanabilir bir yön atanmaktadır. Sonra seçilen yöne atanan kare bir bölge elde edilmektedir ve SURF betimleyicileri bu kare alandan elde edilmektedir. Son olarak da, bu elde edilen öznelikler iki görüntü arasında eşlenmektedir. Bu iki adım aşağıda açıklanmıştır.

- **Yönelim atama**

İlgi noktaları eşleştirilirken yöneline karşı bağımsızlığın sağlanmasında her bir ilgi noktasında tekrarlanabilir bir yön tanımlanmaktadır. Bu amaç ile ilk olarak bir ilgi noktasının  $s$  ölçeğinde olmak üzere, ilgi noktası etrafında  $6s$  yarıçaplı dairesel bir alan içerisindeki komşulukta  $x$  ve  $y$  yönünde Haar dalgacık tepkileri hesaplanmakta ve ilgi noktasında merkezlenmiş bir Gauss'la ( $\sigma = 2s$ ) ağırlıklandırılmaktadır. Ağırlıklandırılan bu tepkiler; düşey tepkiler düşey eksen boyunca, yatay tepkiler yatay eksen boyunca noktalar olarak temsil edilmektedir. Baskın yön, merkez etrafında  $\frac{\pi}{3}$  açısı ile çevreli bir daire dilimi döndürülerek seçilmektedir. Baskın yön, dilimlenmiş yön penceresi içerisindeki bütün cevapların toplamı hesaplanarak tahmin edilmektedir. Pencere içerisindeki düşey ve yatay cevaplar toplanmaktadır. Her bir pozisyonda  $x$  ve  $y$  cevapları dilim içerisinde toplanmakta ve yeni bir vektör oluşturmak için seçilmektedir. En uzun vektör ilgi noktasının yönünü belirlemektedir.

- **Haar dalgacık tepkilerinin toplamına dayanan betimleyiciler**

Betimleyicilerin elde edilmesinde ilk adım ilgi noktaları etrafında merkezlenmiş kare bir pencere oluşturulması (bu pencerenin boyutu  $20s$  olarak alınmıştır) ve yön atama aşamasında seçilen yön doğrultusunda yönlendirilmesidir.

Betimleyici penceresi, daha küçük  $4 \times 4$  kare alt bölgelere düzenli bir şekilde bölünmektedir ve bu da önemli uzaysal bilgiyi korumaktadır. Her alt bölge için,  $5 \times 5$  lik düzenli olarak aralıklandırılmış örnek noktalarda Haar dalgacık tepkileri hesaplanmaktadır. Basitleştirmek için, Haar dalgacık tepkileri yatay yönde  $d_x$ , düşey yönde  $d_y$  olarak adlandırılmıştır. Geometrik deformasyonlar ve konumlandırma hatalarına karşı sağlamlığını artırmak için  $d_x$  ve  $d_y$  tepkileri ilk olarak Gauss ( $\sigma = 3,3s$ ) ile ağırlıklandırılmıştır.

Hesaplanan  $d_x$  ve  $d_y$  tepkileri her alt bölge üzerinden toplandıktan sonra öznitelik vektörünün ilk girdi kümesi oluşturulmaktadır. Yoğunluk değişimlerinin yoğunlaşması hakkında bilgi almak için ayrıca tepkilerin mutlak değerlerinin toplamı da alınmaktadır

( $|d_x|$  ve  $|d_y|$ ). Her alt bölge 4 boyutlu betimleyici vektör  $\mathbf{v}$  aşağıdaki formül ile elde edilmektedir:

$$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (3.16)$$

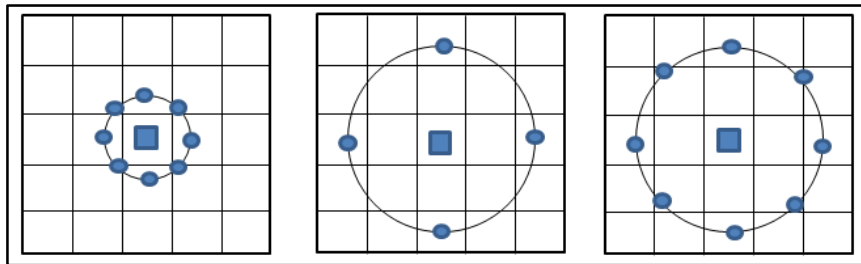
Tüm  $4 \times 4$ 'lük alt bölgeler için bu 4 boyutlu betimleyici vektör birleştirilirse, betimleyici vektör  $4 \times 4 \times 4 = 64$  boyutunda bir vektör haline gelmektedir.

### 3.1.3. Yerel ikili örüntü (LBP)

İlk olarak Ojala vd. (1996) tarafından önerilen Yerel İkili Örüntü (LBP) operatörü yerel komşuluk değerlerine dayalı bir öznitelik çıkarma yöntemidir. LBP operatörü, görüntüdeki her bir piksele bir etiket atamaktadır. Bu etiket merkez pikselin  $R$  komşuluğundaki piksellerle karşılaştırılması sonucu elde edilen ikili bir sayıdır. Her bir piksel, kendisi ve komşuları arasındaki farkın  $u(x)$  basamak fonksiyonu kullanılarak ikilileştirilmesiyle etiketlenmektedir.

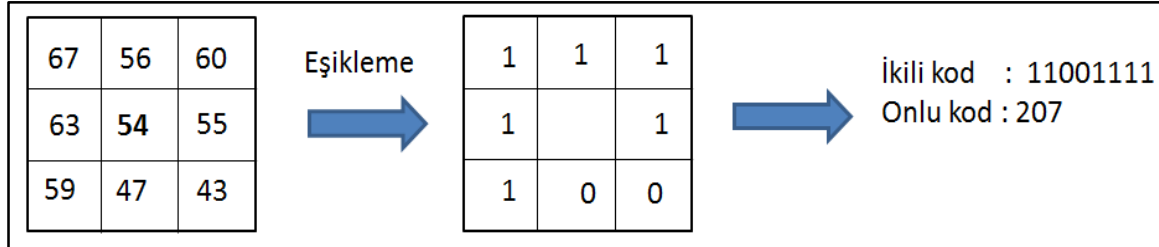
$$LBP_{P,R}(x_c) = \sum_{p=0}^{P-1} u(x_p - x_c) 2^p, \quad u(y) = \begin{cases} 1, & y \geq 0 \\ 0, & y < 0 \end{cases} \quad (3.17)$$

Burada  $x_c$ , LBP etiketi üretilen merkez pikseli,  $x_p$  merkez pikselin komşularını,  $R$  komşuların merkez piksele olan uzaklığını,  $P$  ise işleme giren komşu sayısını göstermektedir. Bu yapı çeşitli dairesel komşulukların kullanılabilceği şekilde genişletilerek farklı ölçeklerde de analiz yapmayı kolaylaştırmıştır. Şekil 3.5'de çeşitli LBP operatörlerine örnekler verilmiştir.



Şekil 3.5 (8,1), (4,2) ve (8,2) LBP operatörleri

LBP operatörü ile piksellerin etiketlenmesine ilişkin bir örnek Şekil 3.6'da verilmiştir.



Şekil 3.6 LBP değerinin hesaplanmasının grafiksel ifadesi. Eşikleme işleminden sonra 8-bitlik LBP kodu oluşturulur. Daha sonra bu kodun onlu sistemdeki karşılığı hesaplanarak merkez pikselin LBP değeri bulunur.

Oluşturulan LBP değerlerinin tümü doku tanımlamada kullanılmamaktadır. Tanımlamada kullanılan düzgün örüntüler, ikili LBP kodunda 0-1 ya da 1-0 geçiş sayısı 2 ya da daha az olanlardır. Örneğin 00000000 ve 11111111 örüntüleri 0 geçişe, 01110000 ve 11001111 örüntüleri 2 geçişe sahip olduğundan düzgündür. Fakat 4 geçişli 11001000 ve 6 geçişli 01010011 örüntüleri düzgün değildir. Yapılan çalışmalarda, düzgün örüntülerin araştırılan doku yüzeylerinde bazen %90'ın üzerinde olduğu görülmüştür (Ojala vd., 1996). Etiketli görüntünün,  $f_l(x, y)$ , histogramı denklem 3.18 ile bulunur:

$$H_i = \sum_{x,y} I\{f_l(x, y) = i\}, i = 0, \dots, n - 1,$$

$$I\{A\} = \begin{cases} 1, & A \text{ doğru ise} \\ 0, & A \text{ yanlış ise.} \end{cases} \quad (3.18)$$

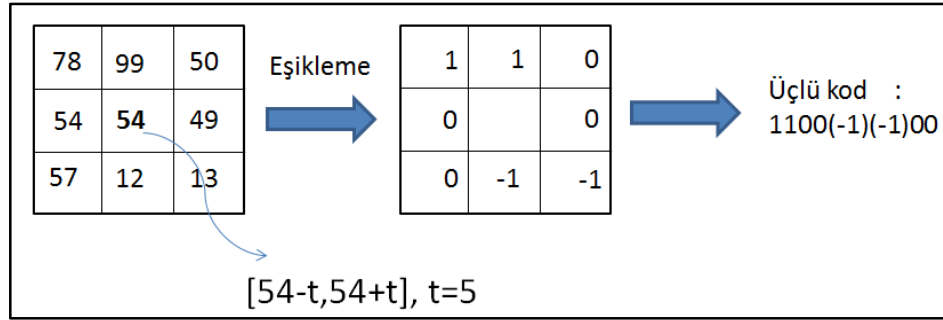
Denklem 3.18'deki  $n$ , LBP operatörü tarafından üretilen farklı etiket sayısıdır. Bu histogram tüm resim üzerindeki kenarlar, benekler, düz alanlar gibi mikro-örüntülerle ilgili bilgi taşımaktadır (Ahonen vd.,2006).

### 3.1.4. Yerel üçlü örüntü (LTP)

Yerel Üçlü Örüntünün (LTP) yerel ikili örüntüyle bir bağıntısı vardır. LBP'nin aksine LTP, eşikleme için 3 değer kullanmaktadır. Gri seviyelerde  $\pm t$  genişliği etrafındaki bölgede  $i_c$  0'a etiketlenmekte, bunun yukarısında olanlar +1 altında olanlar ise -1 olarak etiketlenmektedir. Gösterge  $s(u)$  3 değerli bir fonksiyonla değiştirilmiştir:

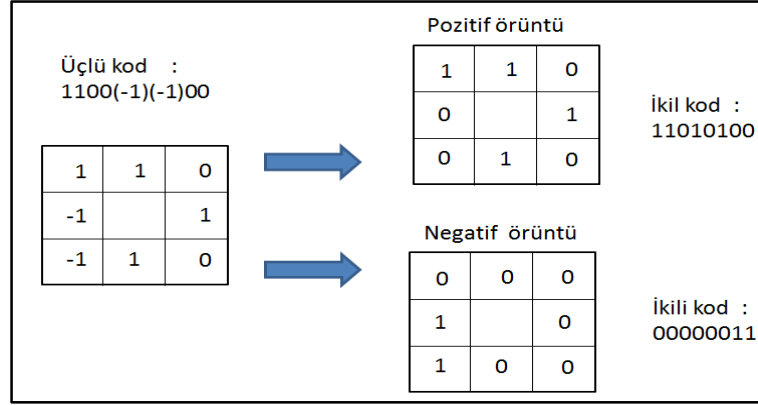
$$s'(u, i_c, t) = \begin{cases} 1, & u \geq i_c + t \\ 0, & |u - i_c| < t \\ -1, & u \leq i_c - t \end{cases} \quad (3.19)$$

Ayrıca ikili LBP kodu da üçlü LTP kodu ile değiştirilmiştir. LTP operatörü ile piksellerin etiketlenmesine ilişkin bir örnek Şekil 3.7'de verilmiştir. Burada eşik değeri 5 seçilmiştir, böylelikle tolerans aralığı  $[49,59]$ dur.



Şekil 3.7 LTP değerinin hesaplanmasının grafiksel ifadesi

Basitlik için, her üçlü kod pozitif ve negatif olarak Şekil 3.8'de gösterildiği gibi 2 parçaya ayrılmıştır. LBP betimleyicileri, ayrı histogram ve benzerlik ölçümleri için 2 kanaldan hesaplanmakta, hesaplama sonunda bunlar birleştirilmektedir (Tan ve Triggs, 2010).



Şekil 3.8 LTP kodunun pozitif ve negatif LBP koduna ayrılmasına ilişkin örnek

### 3.2. Yerel Özniteliklerin Kodlanması

İngeden SIFT, SURF, LBP, LTP gibi betimleyiciler kullanılarak özniteliklerin çıkarılması ile imge erişimi için gerekli olan temel adımlardan ilki gerçekleşmektedir. İkinci adımda ise çıkarılan bu öznitelikler görüntü içeriğini özetleyen tek bir öznitelik vektörü içine kodlanarak öğrenme ve tanıma için uygun bir şekilde getirilmesidir. En iyi bilinen kodlama yöntemi muhtemelen nicemlenmiş vektör betimleyicilerinin histogramından oluşan Kelimeler Çantası Yöntemidir (Csurka, 2004; Jurie ve Triggs, 2005; Fergus, 2005; Harzallah, 2009) Bir diğer kodlama yöntemi ise Fisher Vektör (Perronnin vd., 2010) yöntemidir. Bu kodlamaların ortak amacı vektör nicemele adımındaki bilgi kaybını azaltmaktır.

Tüm kodlama yöntemleri yerel betimleyicilerin bilgilendirici bölgelere bölünmesi fikrine dayanmaktadır. Bu bölgeler görsel kelimeler olarak adlandırılmakta ve görsel kelimelerin toplanmasına da görsel sözlük adı verilmektedir (Chatfield vd., 2011). En yaygın görsel sözlük oluşturma yolu toplama algoritmaları arasında  $k$ -ortalama algoritmasıdır. Bu algoritma  $N$  tane eğitim betimleyicisi  $x_1, \dots, x_N \in R^D$  verildiğinde,  $k$  tane kümeye bölünmesi gerekmektedir. Burada amaç her bir noktayı bir kümeye atamaktır.  $k$ -ortalama toplama yöntemi veri noktalarının kümeye olan uzaklıklarını en aza indirgenecek şekilde kümelerin konumlarını bulmayı amaçlamaktadır.  $k$ -ortalama toplama yöntemi aşağıdaki denklemle çözülmektedir:

$$\operatorname{argmin}_c \sum_{i=1}^k \sum_{x \in c_i} d(x, \mu_i) = \operatorname{argmin}_c \sum_{i=1}^k \sum_{x \in c_i} \|x_i - \mu_{qi}\|^2 \quad (3.20)$$

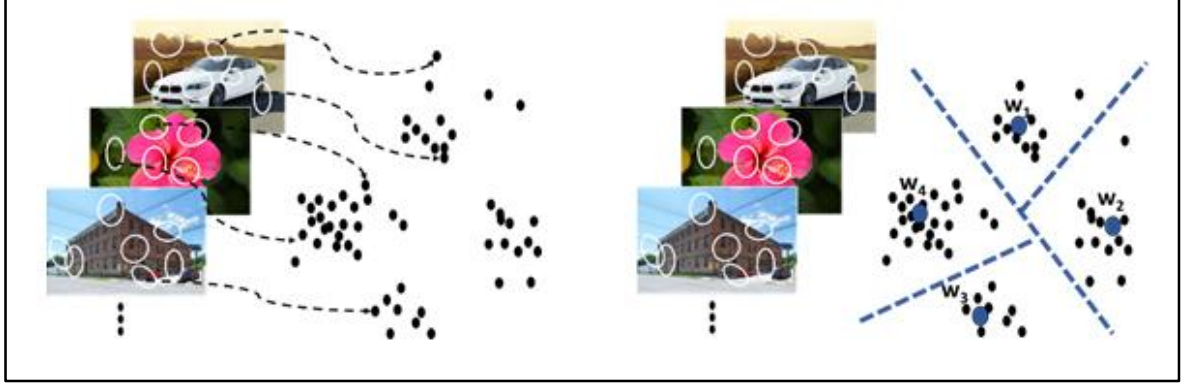
Denklem 3.20'de geçen  $c_i$  küme  $i$ 'ye ait noktalar kümesini ifade etmektedir ve  $k$ -ortalama toplama yöntemi Öklid uzaklığın karesini  $d(x, \mu_i) = \|x_i - \mu_{qi}\|^2$  kullanmaktadır.

### 3.2.1. Kelimeler çantası tabanlı imge betimleme

Literatürde ilk olarak imge sınıflandırma amacıyla önerilen bu yöntem, çok kısa bir zamanda başarılı sonuçlar elde edilmesiyle birlikte popüler hale gelmiştir. Kelimeler Çantası modeli nesne algılama, görüntü alma ve robotlar için görsel nesne konum bulma gibi pek çok alanda uygulanmıştır (Csurka, 2004; Jurie ve Triggs, 2005; Fergus, 2005; Harzallah, 2009). Kelimeler çantası tabanlı imge betimleme her bir imgeyi sıralı temsili yamaların birleşimi olarak derlemektedir. Görsel kelimeler, eğitim setindeki imgelerden çıkarılan özniteliklerin kümelenmesiyle oluşturulmuştur. Bu nedenle yöntem imgelerden belirli teknikler kullanılarak birçok yamanın seçimini gerektirmektedir. Daha sonra bu yamalar SIFT, LBP vb. gibi betimleyiciler kullanılarak sabit boyutlu yama vektörlerine dönüştürülmektedir. Bir sonraki aşama ise bu yama vektörlerinin nicemlenerek  $k$  tane temsili yama vektörlerinin belirlenmesidir. Son olarak imgelerden gelen yamaların temsili yamalara olan uzaklıklarına bakılarak her bir yama temsili yamaya atanmakta ve imgeler bu atamalardan oluşturulan  $k$  boyutlu histogramlar olarak betimlenmektedir (Çevikalp vd., 2013).

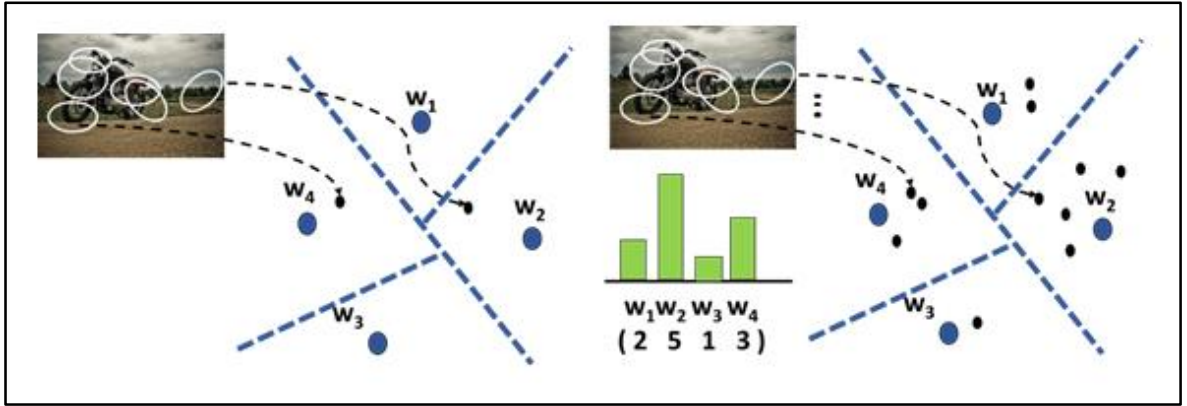
İmgelerden yamaları seçmek için belirgin nokta bulma algoritmaları kullanıldığı gibi, yamalar farklı pozisyonlarda ve ölçeklerde rassal olarak ya da sık aralıklarla da seçilebilmektedir (Jurie, 2005; Nowak, 2006). Bazı durumlarda yamalar imge bölütleme (image segmentation) algoritmasının döndürdüğü bölütler etrafından seçilmektedirler (Koniusz, 2010). İmgeden seçilen yamalar SIFT, LBP gibi betimleyiciler tarafından sabit boyutlu vektörlere dönüştürüldükten sonra yama vektörleri nicemeleme işlemine tabi tutulmaktadır. Bu amaç için en çok kullanılan yöntem  $k$ -ortalama toplama algoritmasıdır. Bu toplama algoritmasının döndürdüğü merkezler temsili yama vektörleri (temsili yama vektörleri BoW modelinde words-kelimelere karşılık gelmektedir) olarak kullanılmaktadır. Şekil 3.9'da gösterildiği üzere imgelerden gelen yama vektörlerinin

temsili yama vektörlerine olan uzaklıklarına bakılarak atamalar yapılmakta ve imgeleri temsil eden histogram vektörleri belirlenmektedir.



(a)

(b)



(c)

(d)

Şekil 3.9 Kelimeler çantası temelli imge betimlemenin şematik gösterimi. (a) Eğitim setindeki imgelerin öznelikleri çıkarılarak öznelik uzayı doldurulmuştur. Beyaz elipsler yerel öznelik bölgelerini göstermekte, siyah noktalar ise öznelik uzayında çıkarılan betimlenmiş noktaları belirtmektedir. (b) Çıkarılan öznelikler görsel kelimeleri elde etmek için kümelenmiştir. Görsel kelimeler küme merkezleridir ve büyük mavi dairelerle gösterilmiştir. Noktalı mavi çizgiler, seçilen küme merkezlerine dayalı Voronoi hücrelerini ifade etmektedir. (c) Verilen yeni imgenin her bir özneliği için en yakın görsel kelime belirlenmiş ve her bir öznelik belirlenen görsel kelimeye atanmıştır (şekilde  $w_i, i = 1, \dots, 4$  olarak gösterilen 4 tane görsel kelime vardır). (d) Son olarak imge, atamaları yansıtan histogram vektörü ile temsil edilmektedir (Grauman ve Leibe, 2009).

### 3.2.2. Fisher vektör ile imge betimleme

Fisher vektör gösterimine; her bir imgeden seçilen yamaların SIFT (Lowe, 2004), LTP gibi betimleyicilerle özneliklerinin çıkarılmasıyla başlanılır. Fisher vektör



kodlaması; büyük bir vektör kümesini (çıkarılan SIFT öznitelikleri gibi) yüksek boyutlu vektör gösterimi içine toplar. Genel olarak, bu işlem özniteliklere “Gauss Karışım Modeli (GKM)” gibi bir parametrik generatif model uyarlanarak ve daha sonra parametrelerine göre modelin log-olabilirliği türevi kodlanarak yapılır (Jaakkola ve Haussler, 1998). GKM diyagonal kovaryanslarla eğitilir ve sadece Gauss ortalaması ve varyansı ile ilgili türevleri dikkate alınır. Öznitelikler ve GKM merkezlerinin her biri arasındaki ilk ve ikinci dereceden farkların ortalamasını yakalayan bir gösterim elde edilmiş olur. Bir Fisher vektör  $G_\lambda^X$ , farkların kümelenmesiyle elde edilir.

### 3.2.2.1. Fisher çekirdek yapısı

$X = \{x_t, t = 1, \dots, T\}$ , imgeden çıkarılan  $T$  tane yerel betimleyici kümesini ifade eder.  $X$  oluşturma sürecinin,  $\lambda$  parametrelili olasılık yoğunluk fonksiyonu  $u_\lambda$  tarafından modellenilebilir olduğu varsayılmıştır.  $X$ , gradyen vektörü ile ifade edilebilir (Jaakkola ve Haussler, 1998):

$$G_\lambda^X = \nabla_\lambda \log u_\lambda(X|\lambda) \quad (3.21)$$

Log-olabilirliğinin gradyeni, parametrelerin oluşturma sürecine katkısını açıklar. Bu vektörün boyutluluğu sadece  $\lambda$  içindeki parametrelerin sayısına bağlıdır. Bu gradyen üzerindeki doğal çekirdek şöyle ifade edilir:

$$K(X, Y) = G_\lambda^{X^T} F_\lambda^{-1} G_\lambda^Y \quad (3.22)$$

Burada  $F_\lambda$ ,  $u_\lambda$  ‘nın Fisher bilgi matrisidir:

$$F_\lambda = E_{x \sim p} [\nabla_\lambda \log u_\lambda(x|\lambda) \nabla_\lambda \log u_\lambda(x|\lambda)'] \quad (3.23)$$

$F_\lambda$ , simetrik ve pozitif-kesin olduğu gibi Cholesky ayrışmasına  $F_\lambda = L_\lambda^T L_\lambda$  da sahiptir.  $K(X, Y)$ , normalize edilmiş  $G_\lambda$  vektörleri arasındaki iç çarpım gibi yazılabilir:

$$G_\lambda^X = L_\lambda G_\lambda^X \quad (3.24)$$

$G_\lambda^X$ ,  $X$ 'in Fisher vektörünü (FV) temsil eder.

### 3.2.2.2. Fisher vektör ile imge gösterimi

Perronin ve Dance'ın (2007) çalışmasına göre GKM olarak  $u_\lambda$  seçilir:  $u_\lambda = \sum_{i=1}^K w_i u_i(x)$ . Gauss  $u_i$ 'nin  $w_i, \mu_i \sigma_i$ , sırasıyla karışım ağırlığı, ortalaması ve diyagonal varyans matrisleri olmak üzere  $\lambda = \{w_i, \mu_i \sigma_i, i = 1, \dots, K\}$  ifade edilmektedir. Bu  $u_\lambda$  modeli görsel kelime gibi anlaşılabilir. GKM  $u_\lambda$ , çok sayıda imge üzerinde maksimum olasılık ile eğitilebilir.  $x_t$ ,  $u_\lambda$  kullanılarak bağımsız olarak oluşturulabilir:

$$G_\lambda^X = \frac{1}{T} \sum_{t=1}^T \nabla_\lambda \log u_\lambda(x_t) \quad (3.25)$$

Perronin ve Dance'ın (2007) çalışmasındaki Fisher bilgi matrisinin, diyagonal kapalı form yaklaşımı kullanılmaktadır.  $\gamma_t, x_t$  betimleyicisinin  $i$ . Gauss soft atamasıdır:

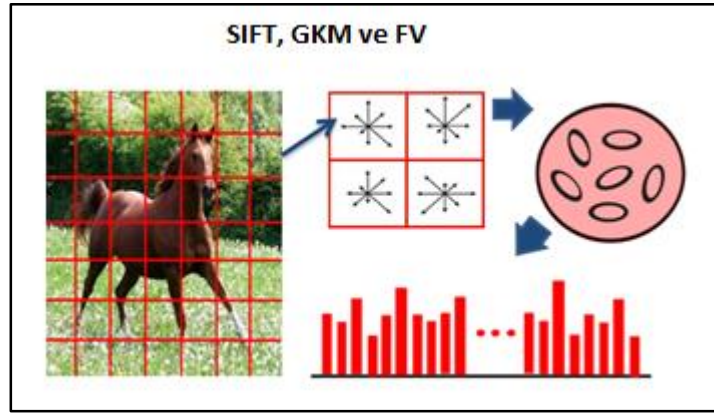
$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^K w_j u_j(x_t)} \quad (3.26)$$

$G_{\mu,i}^X$ ,  $i$ . Gaussun ortalaması  $\mu_i$  ile ilgili  $d$ - boyutlu gradyendir,  $G_{\sigma,i}^X$  de benzer şekildedir.  $u_t$  tarafından bağımsız olarak  $x_t$ 'nin oluşturulduğu varsayılırsa, standart matematiksel türev elde edilebilir:

$$G_{\mu,i}^X = \frac{1}{T \sqrt{w_i}} \sum_{t=1}^T \gamma_t(i) \left( \frac{x_t - \mu_i}{\sigma_i} \right) \quad (3.27)$$

$$G_{\sigma,i}^X = \frac{1}{T \sqrt{2w_i}} \sum_{t=1}^T \gamma_t(i) \left[ \frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right] \quad (3.28)$$

Fisher vektörünün son hali  $G_\lambda^X$ ,  $G_{\mu,i}^X$  ve  $G_{\sigma,i}^X$ 'nin  $i = 1, \dots, K$  için birleşimidir. Farkların:  $G_\lambda^X = [G_{\mu,1}^X, G_{\sigma,1}^X, \dots, G_{\mu,K}^X, G_{\sigma,K}^X]$  kümelenmesiyle bir Fisher vektör  $G_\lambda^X$  elde edilmiştir ve toplamda  $2Kd$ -boyutludur. Burada  $K$ , GKM'deki Gauss sayısını ve  $d$ 'de öznitelik vektörlerinin Temel Bileşen Analizi (TBA) işlemi sonrası boyutunu göstermektedir Kelimeler Çantası yöntemi ile karşılaştırıldığında, aynı uzunlukta imza elde etmek için  $d$  kat daha az görsel kelime gereklidir. Deneysel sonuçlara göre güzel sonuçlar az sayıda görsel kelime  $K$  ile elde edilebilir. Fisher vektör ile imge betimlemenin şekilsel gösterimi Şekil 3.10'da gösterilmiştir.



Şekil 3.10 Fisher vektör ile imge betimleme.

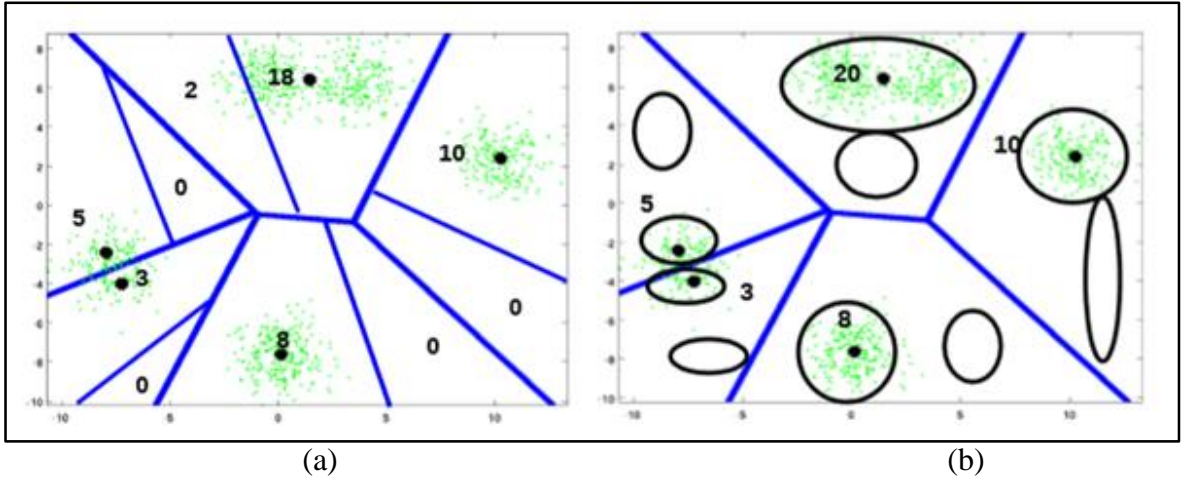
### 3.2.3. Fisher vektör ile kelimeler çantası yönteminin karşılaştırması

Verbek (2012) tarafından Fisher Vektör ile Kelimeler Çantası Yöntemi arasındaki benzerlik ve farklılıklar aşağıdaki gibi belirtilmiştir:

1. Hem Fisher vektör hem de kelimeler çantası yöntemi görsel sözlüğe dayanmaktadır.
2. Kümeleme için Fisher vektör Gauss karışım modelini, kelimeler çantası yöntemi ise  $k$ -ortalama kümelemeyi kullanmaktadır.
3. Verilen görsel kelime sayısı için Fisher vektör, kelimeler çantası yönteminden daha fazla imge imzası çıkarmaktadır.
4. Kelimeler çantası yöntemi doğrusal olmayan sınıflandırıcıları gerektirirken, Fisher vektörler doğrusal sınıflandırıcı kullanılarak sınıflandırılabilir. Sonuç olarak, sınıflandırma sürecinde Fisher vektörler daha az hesaplama karmaşıklığı gerektirmektedir.

5. Kelimeler çantası histogramı her kelimeye atanan yamaların sayısını saklamaktadır. Gösterimi geliştirmek için daha fazla kelimeye ihtiyaç duymaktadır. Fakat bu da doğrudan hesaplama maliyetini artırmaktadır ve birçok kez boş yere fazlalığa neden olmaktadır. Bunun yerine, Fisher vektör her hücrede noktaların ortalaması ve varyansını saklamaktadır. Aynı sayıda görsel kelimeyle daha çok bilgi saklamakta ve hesaplama süresinde belirgin olarak bir artış olmamaktadır. Yüksek boyutlu öznelik vektörüne yol açmaktadır.

Kelimeler Çantası yönteminde  $k$ -ortalama kümeleme ve Fisher Vektör yönteminde gauss karışım modeli ile kümeleme gösterimi Şekil 3.11’de gösterilmektedir.



Şekil 3.11 (a) Kelimeler Çantası yönteminde  $k$ -ortalama kümeleme gösterimi. (b) Fisher vektör yönteminde gauss karışım modeli ile kümeleme gösterimi (Jakob, 2012).

#### 4. YEREL DUYARLI ÖZETLEME (LOCALITY SENSITIVE HASHING)

İmge erişiminde, veri tabanındaki imge sayısının çok fazla olduğu durumlarda hem işlem yükü hem de saklama yükü artmaktadır. Bu sebeple imge histogramlarını daha etkin bir biçimde hafızada tutabilmek amacıyla özetleme işlemi yapılmaktadır. Özetleme yönteminde her bir imge  $b$ -bitten oluşan ve özet vektörü adı verilen bir vektör ile ifade edilmektedir. Özet vektörleri hafızada daha az yer tutacak şekilde saklanmaktadır. Ayrıca bit düzeyinde işlemler çok hızlı bir şekilde gerçekleştiğinden özetleme yöntemi kullanıldığında sorgu imgesine benzer imgeleri bulma işlemi oldukça hızlanmaktadır. Özetleme işleminde amaç  $k$  boyutlu örnek uzayında imge histogramlarının arasındaki uzaklıkları koruyan özet kodları üretmektir. Bu sebeple yöntem genellikle Yerel Duyarlı Özetleme (YDÖ) olarak adlandırılmaktadır. Bu amaçla önerilen ilk yöntemlerde rassal olarak izdüşüm vektörleri belirlenerek imge vektörlerinin bunların üzerlerine olan izdüşümleri bulunmuş daha sonra yine rassal olarak belirlenen eşik değerleri kullanılarak, imgelerin izdüşümleri ikili kodlar ile kodlanmıştır (Gionis vd., 1999; Raginsky ve Lazebnik, 2009). Salakhutdinov ve Hinton (2009) yazarlar özet kodlarını üretmek için rassal izdüşüm vektörleri yerine Restricted Boltzman Machines kullanmışlar ve bu yöntemi metin erişimi için kullanmışlardır. Kulis ve Grauman (2009) özetleme yöntemini kernel fonksiyonlarıyla da çalışacak şekilde geliştirmişlerdir. Fakat bu yöntemde  $N \times N$  ( $N$  -imge histogram vektör sayısı) boyutlu bir kernel matrisine ihtiyaç duyulduğundan veri sayısının fazla olduğu durumlarda bu yöntemi doğrudan uygulamak mümkün değildir. Bu sebeple yazarlar, verinin arasından oldukça az sayıda örnek alarak sadece bu örnekleri kullanarak yöntemi uygulamışlardır. Bu sebeple yöntem birçok veriyi etkili bir şekilde kullanamamaktadır. Ayrıca sorgu imgesine ait özet kodunu oluşturmak için seçilen örnekleri kullanarak kernel fonksiyonun gerçekleşmesi gerekir ve bu da test hızını oldukça kötü etkilemektedir. Weiss ve diğerleri (2009) de özet kodlarını üretmek için spektral topklamada kullanılan formüle benzer bir formül kullanılarak “Spectral Hashing” adını verdikleri yöntemi geliştirmişlerdir. Bu yöntemde ilk olarak giriş örnek uzayındaki imge vektörlerinin birbirlerine olan uzaklıkları kullanılarak 0 ile 1 arasında değişen benzerlik oranları belirlenmiş ve bu benzerlikleri koruyan özet kodları üretilmeye çalışılmıştır. Yöntem  $N \times N$  boyutunda ikili örnekler arasındaki benzerliklerin sayısal değerlerini içeren bir benzerlik matrisi (affinity matrix) kullanıldığından bu yöntemin veri sayısı fazla olan

veri tabanlarında kullanılması mümkün değildir. Bu sorunu çözmek için bir yaklaşım önerilmiştir, fakat bu yaklaşım imge vektörlerinin dağılımının Gauss olması durumunda geçerlidir. Daha da kötüsü yöntem imge verileri üzerinde Temel Bileşen Analizi-TBA (Principal Component Analysis) uygulanmasını gerektirmektedir ve TBA uygulanabilmesi için veri uzay boyutunun küçük olması gerekir. Fakat kelime çantası temelli betimlemede imge histogram vektörlerinin boyutlarının 100K civarında olması sebebiyle bu yaklaşımı kullanmak oldukça zordur. Bu sebeple yazarlar, yöntemlerini sadece imge uzay boyutu çok düşük ve gerçekçi olmayan ( $32 \times 32$  boyutlu imgeler) 80 milyonluk “Tiny Images” veri tabanında uygulanmışlardır (yöntem bu küçük boyutlu imgelerden elde edilen ve yine boyutları düşük olan Gist betimleyiciler kullanılarak elde edilen imge vektörlerine uygulanmıştır ve imge vektörlerinin boyutu 64 gibi oldukça küçük bir sayıdır). Liu ve diğerleri de (2011) spektral yaklaşımı kullanmışlar ve boyut sorununu çözmek için ilk önce en yakın merkez toplama algoritmasını kullanarak sınırlı sayıda ( $p$  diyelim  $p \ll N$ ) “anchor” adını verdikleri noktalar belirlemişler ve  $N \times N$  boyutlu benzerlik matrisi yerine daha küçük boyutlu  $p \times p$  benzerlik matrisini kullanmışlardır.

Özetleme işleminde amaç  $k$  boyutlu örnek uzayında imge histogramlarının arasındaki uzaklıkları koruyan özet kodları üretmedir ve bu neden ile kullanılan yöntem genellikle Yerel Duyarlı Özetleme (YDÖ) olarak adlandırılmaktadır. Yerel Duyarlı Özetleme yönteminin ana fikri; noktaları çeşitli özet fonksiyonları kullanarak özetlemektir. Bu işlem yapılırken her bir fonksiyon için, birbirine yakın olan nesnelere çarpışma olasılığının, birbirine uzak olan nesnelere daha yüksek olduğu göz önüne alınmalıdır. Sonrasında sorgu noktası özetlenerek, bu noktayı içeren kovalarda saklanan öğeler alınarak yakın komşular belirlenebilir (Datar vd, 2004).

Yerel Duyarlı Özetleme öklid uzayında  $l_2$  yüksek boyutlu yakın komşu problemine rastgele çözüm sağlayan bir yöntemdir.  $R$ -yakın komşu problemi aşağıdaki gibi tanımlanmaktadır. Noktalar kümesi  $P \subset \mathbb{R}^D$  ve yarıçap  $R > 0$  verildiğinde, aşağıdaki sorgulara yanıt veren bir veri yapısı oluşmaktadır: sorgu noktası  $q$  için,  $p \in P, \|q - p\|_2 \leq R$  olacak şekilde tüm noktaları bulur ( $\|q - p\|_2$   $q$  ve  $p$  arasındaki Öklid uzaklıktır). LSH,  $(R, 1 - \delta)$  yakın komşu problemi olarak adlandırılan bu problemin rasgele sürümünü çözer. Bu durumda, her  $p$  noktası  $\|q - p\|_2 < R$  eşitsizliğini sağlayarak, en az  $1 - \delta$

olasılığıyla rapor edilmelidir.(Andoni ve Indyk, 2005). Bu bölümde Datar vd. (2004) tarafından yapılan Yerel Duyarlı Özetleme algoritmasının detayları anlatılacaktır.

#### 4.1. Gösterim

$d$  boyutlu gerçekte uzayda  $\mathbb{R}^D$ ,  $l_p$  normu altında  $l_p^d$  kullanılmaktadır. Herhangi bir nokta  $v \in \mathbb{R}^D$  için,  $\|v\|_p$  gösterimi  $v$  vektörünün  $l_p$  normunu ifade etmektedir:

$$\|v\|_p = \left( \sum_{i=1}^d v_i^p \right)^{1/p} \quad (4.1)$$

Özellikle  $\|v\| = \|v\|_2$  eşitliği Öklid normdur.  $P$  veri kümesinin  $\mathbb{R}^D$ 'nin sınırlı altkümesi olduğunu ve  $n = |P|$  olduğunu varsayalım. Bir nokta  $q$  genellikle sorgu noktası olacaktır ve sorgu noktası  $\mathbb{R}^D$ 'den herhangi bir noktadır.  $v$  ve  $u$  noktaları  $P$  veri kümesinde bulunan bazı noktalardır.  $v$  merkezli  $r$  yarıçaplı küre  $B(v, r)$  ile ifade edilmektedir. Her  $q$  sorgu noktası için, eğer  $v \in B(q, R)$  ise  $v$ ,  $R$ -yakın komşu olarak adlandırılır.

#### 4.2. Genel Yerel Duyarlı Özetleme

Indyk ve Motwani'nin (1998) çalışması  $(R, c)$ -yakın komşu problemi olan Yerel Duyarlı Özetlemeyi çözmek için önemli bir tekniktir. Uzaklık ölçütü  $D$  olan noktalar kümesinin  $S$  tanım kümesi için, YDÖ şu şekilde ifade edilmektedir (Andoni ve Indyk, 2005):

**Tanım 1.** Eğer fonksiyon  $p(t) = P_{rH} = [h(q) = h(v)] : \|q - v\| = t$  herhangi  $q$  için kesinlikle azalıyorsa,  $H = \{h: S \rightarrow U\}$  kümesi, yerel duyarlı olarak adlandırılır. Yani  $q$  ve  $v$  noktalarının çarpışma olasılığı aralarındaki mesafe ile azalmaktadır.

Böylece, eğer  $q, v, u$  noktaları  $v \in B(q, R)$  ve  $u \notin B(q, R)$  ise  $p(\|q - v\|) > p(\|q - u\|)$  olur.  $U$  tanım kümesi içinde  $P$ 'den noktalar özetlenir ve sonra sorgu sırasında  $q$ 'nun özeti hesaplanır ve sadece  $q$  ile çarpışan noktalar dikkate alınır.

Ancak, istenen çalışma süresini elde etmek için,  $[0, R]$  ve  $[R, \infty]$  aralıkları için çarpışma olasılıkları arasındaki boşluğu yükseltmek gereklidir. Bu amaç için çeşitli özet fonksiyonları  $h \in H$  birleştirilmiştir.  $G = \{g: S \rightarrow U^k\}$  fonksiyon kümesi  $g(v) = (h_1(v), \dots, h_k(v))$ ,  $h_i \in H$  olacak şekilde tanımlanmıştır.  $L$  tamsayısı için, algoritma tarafından bağımsız ve eşit olarak rastgele  $L$  tane fonksiyon  $g_1, \dots, g_L$   $G$ 'den seçilmiştir. Önişleme sırasında, her  $v \in P$  (giriş nokta kümesi) kovalarda  $g_j(v)$ ,  $j = 1, \dots, L$  olacak şekilde saklanmaktadır. Toplam kova sayısı büyük olabilir ancak algoritma sadece boş olmayan kovaları özetleme için muhafaza etmektedir.

Bir sorgu  $q$  işlemi için, algoritma tüm kovaları  $g_1(q), \dots, g_L(q)$  arar. Kovada bulunan her  $v$  noktası için, algoritma  $q$  ve  $v$  arasındaki mesafeyi hesaplar ve  $\|q - v\| \leq R$  ise  $v$  noktası bildirilir ( $v, R$ -yakın komşudur).

### 4.3. $l_p$ Normu için Yerel Duyarlı Özetleme

Yerel Duyarlı Özetleme yöntemi Datar vd. (2004)'in önerdiği  $p$ -kararlı dağılımlarına dayanmaktadır ve tüm  $p \in (0, 2]$  için çalışmaktadır.

#### 4.3.1. $p$ -kararlı dağılımları

Kararlı dağılımlar (Zolotarev, 1986) aynı şekilde dağıtılan bağımsız değişkenlerin normalize edilmiş toplamlarının sınırı olarak tanımlanabilir. Kararlı bir dağılımın en iyi bilinen örneği Gauss (veya normal) dağılımıdır.

**Kararlı dağılımlar:**  $\mathbb{R}$  üzerine  $D$  dağılımı  $p$ -kararlı olarak adlandırılır, eğer herhangi  $n$  gerçektek sayı için  $v_1, \dots, v_n$  ve  $D$  dağılımıyla  $X_1, \dots, X_n$  değişkenleri olacak şekilde  $p \geq 0$  mevcut ise  $\sum_i v_i X_i$  rasgele değişkenler,  $D$  dağılımıyla  $X$  rasgele değişken olduğunda  $(\sum_i |v_i|^p)^{1/p}$  değişkeniyle aynı dağılıma sahiptir.

Bilindiği üzere (Zolotarev, 1986), kararlı dağılımlar herhangi bir  $p \in (0, 2]$  için mevcuttur. Özellikle:

- *Cauchy dağılımı*  $D_C$ , yoğunluk fonksiyonu  $c(x) = \frac{1}{2} \frac{1}{1+x^2}$  olarak tanımlanır.



- *Gauss(normal) dağılımı*  $D_G$ , yoğunluk fonksiyonu  $g(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$  olarak tanımlanır.

#### 4.3.2. Özet kümesi

Datar vd. (2004) tarafından önerilen Yerel Duyarlı Özetleme  $p$ -kararlı dağılımları aşağıdaki gibi kullanır: her  $v$  vektörüne özet değeri atamak için iç çarpım  $(a \cdot v)$  hesaplanır. Her özet fonksiyonu  $h_{a,b}(v): \mathbb{R}^d \rightarrow \mathbb{Z}$ ,  $d$  boyutlu  $v$  vektörü tamsayı kümesi üzerine haritalanır. Her özet fonksiyonu rasgele  $a$  ve  $b$  seçimiyle endekslenir. Burada  $a$ ,  $p$ -kararlı dağılımdan bağımsız olarak seçilen girdilerle  $d$  boyutlu vektördür ve  $b$  ise  $[0, w]$  aralığından seçilen gerçek bir sayıdır. Sabit  $a$  ve  $b$  için, özet fonksiyonu aşağıdaki denklemde verilmiştir:

$$h_{a,b}(v) = \left\lfloor \frac{(a \cdot v) + b}{w} \right\rfloor \quad (4.2)$$

Özet fonksiyonu aşağıdaki gibi açıklanabilir. İç çarpım  $(a \cdot v)$  ile her vektörün bir doğru üzerine izdüşümü alınır. Her iki vektör  $(v_1, v_2)$  için izdüşümleri arası uzaklık  $(a \cdot v_1 - a \cdot v_2)$ ,  $\|v_1 - v_2\|_p X$  olarak dağıtılır. Burada  $X$  ile ifade edilen  $p$ -kararlı dağılımdır. Eğer doğru eş genişlikli uygun  $w$  büyüklüğünde parçalara ayrılırsa ve izdüşümünün hangi bölüme düştüğü taban alınarak vektörlere özet değerler atanırsa, yukarıda açıklanan anlamda bu özet fonksiyonu yereldir.

İki vektörün  $v_1, v_2$  özet fonksiyonu altında çarpışma olasılığı hesaplanabilir.  $f_p(t)$   $p$ -kararlı dağılımının mutlak değerinin olasılık yoğunluk fonksiyonu olarak belirtilir. İki vektör  $v_1, v_2$  için,  $c = \|v_1 - v_2\|_p$  olduğu varsayılır. Girdileri  $p$ -kararlı dağılımından çekilmiş  $a$  rasgele vektörü için,  $a \cdot v_1 - a \cdot v_2, cX$  olarak dağıtılmıştır, burada  $X$   $p$ -kararlı dağılımından çekilmiş rasgele bir değişkendir.  $b, [0, w]$ 'den eşit olarak çekilmişse çarpışma olasılığı aşağıdaki eşitlik ile ifade edilebilir:

$$p(c) = Pr_{a,b}[h_{a,b}(v_1) = h_{a,b}(v_2)] = \int_0^w \frac{1}{c} f_p\left(\frac{t}{c}\right) \left(1 - \frac{t}{w}\right) dt \quad (4.3)$$

Sabit  $w$  parametresi için  $p(c)$  çarpışma olasılığı  $c = \|v_1 - v_2\|_p$  ile tanım 1'i sağlayarak tekdüze azalır.

#### 4.4. Yerel Duyarlı Özetlemenin Parametreleri

Üç ana parametre Yerel Duyarlı Özetleme algoritmasının performansını etkiler: her bir özet değerinin izdüşüm sayısı ( $k$ ), özet tablosu sayısı ( $L$ ) ve izdüşüm genişliği ( $w$ ). Problem formülasyonundan, yakın komşu en az  $1 - \delta$  olasılıkla bildirilmiştir, buradan  $k$  ve  $L$  için gerekli koşul elde edilebilir. Sorgu noktası  $q$  ve yakın komşu  $v \in B(q, R)$  olduğu varsayılmıştır.  $p_1 = p(1) = P(R)$  kabul edilmiştir. Ayrıca  $Pr_{g \in G}[g(q) = g(v)] \geq p_1^k$  dir. Böylece  $q$  ve  $v$ , en çok  $(1 - p_1^k)^L$  olasılığıyla tüm  $L$  fonksiyonları  $g_i$  için çarpışma başarısız olur.  $q$  noktasının bazı  $g_i$  fonksiyonu üzerinde  $v$  ile çarpışması için  $1 - (1 - p_1^k)^L \geq 1 - \delta$  eşitsizliğini diğer bir ifadeyle  $L \geq \frac{\log \delta}{\log(1 - p_1^k)}$  eşitsizliğini sağlaması gereklidir. Böylece  $L = \left\lceil \frac{\log \delta}{\log(1 - p_1^k)} \right\rceil$  seçilebilir Çalışma süresi  $L$  ile artmaktadır.

Sabit  $k$  değeri ve  $L = L(k)$  için, sorgu süresi iki ifadeye ayrılmaktadır. İlki sorgu noktası  $q$  için özet tablosundan dönen kovaları  $g_i(v)$  hesaplamak için geçen süre  $T_g = O(dkL)$  ile ifade edilir. Diğeri ise dönen tüm kovalarda çarpışan tüm noktalara olan uzaklığının hesaplanma süresi  $T_c = O(d \cdot \#çarpışma)$  ile ifade edilir.  $\#çarpışma$  kovalarda  $g_1(q), \dots, g_L(q)$  çarpışan nokta sayısıdır; beklenen  $\#çarpışma$  değeri  $E[\#çarpışma] = L \cdot \sum_{v \in P} p^k(\|q - v\|)$  ile ifade edilir.

$T_g$ ,  $k$  fonksiyonu olarak artarken,  $T_c$ ,  $k$  fonksiyonu olarak azalmaktadır. Diğer bir nedenle  $k$ 'nin yüksek değerleri “uzak” ve “yakın” noktaların çarpışma olasılığı arasındaki boşluğu artırırken, uygun  $L$  değeri için uzak noktaların çarpışma olasılığını azaltır. Böylece verilen sorgu noktası  $q$  için,  $T_c + T_g$  toplamını küçükleyen en uygun  $k$  değeri bulunur (Andoni ve Indyk, 2005).

$k$  parametresi özet değerlerini hesaplamada harcanan zaman ve hatalı pozitifleri ayıklamada harcanan zaman arasında ikilem oluşturur. Örneğin, sorgu noktası ve diğer

noktalar arasındaki uzaklık hesaplamada, büyük  $k$  değeri özet hesaplama sayısını artırmaktadır. İzdüşüm genişliğini ( $w$ ) azaltma herhangi iki noktanın çarpışma olasılığını azaltmaktadır. Böylece  $k$  parametresini artırmakla aynı etkiyi yapar. Sonuç olarak,  $r$  mümkün olduğunca küçük seçilmelidir ve bu şekilde izdüşüm sayısının azaltılması gerekir. (Datar vd., 2004).

#### 4.5. Yerel Duyarlı Özetleme Algoritması

Yerel Duyarlı Özetleme algoritması iki aşamadan oluşur: verinin ön işlemeden geçirilmesi ve sorgu noktasının komşularını arama (Şekil 4.1 ve Şekil 4.2):

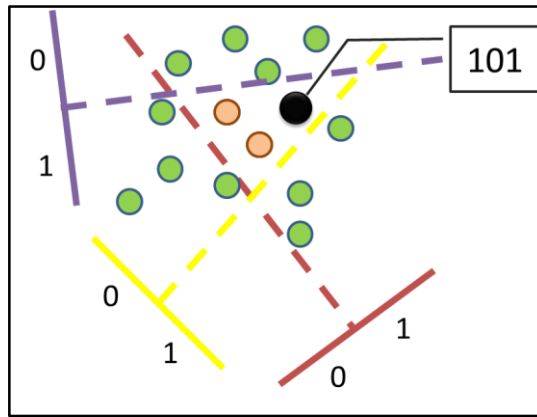
##### Ön İşleme

1. Rasgele  $H$  kümesinden seçilen  $h_{1,j}, h_{2,j}, \dots, h_{k,j}$  değerlerinden oluşan  $g_j = (h_{1,j}, h_{2,j}, \dots, h_{k,j})$ ,  $j = 1, \dots, L$  fonksiyonundan  $L$  tane seçilir.
2.  $L$  tane özet tablosu oluşturulur. Her  $j = 1, \dots, L$  için  $j$ . özet tablosu veri tabanındaki noktaların  $g_j$  fonksiyonu kullanılarak özetlenir.

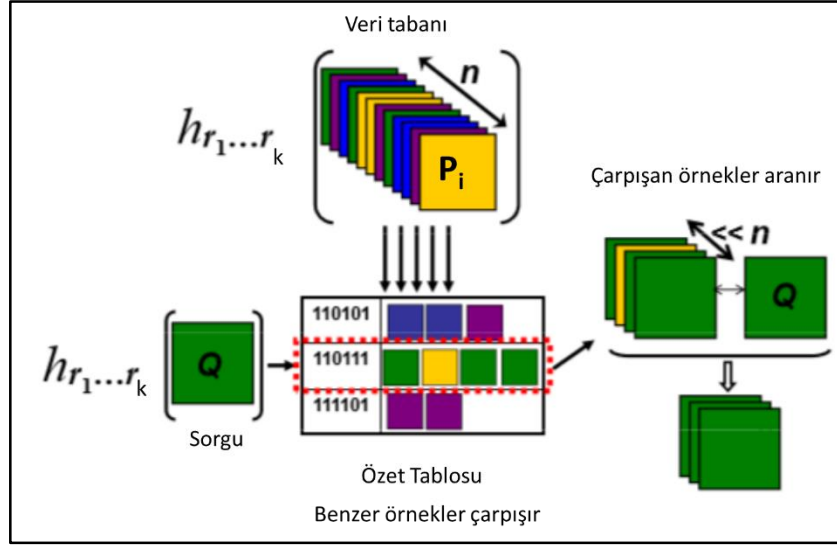
##### Sorgu noktasının ( $q$ ) komşularını arama

Her  $j = 1, \dots, L$  için,

1.  $j$ . özet tablosu içinde  $g_j(q)$  kovanından noktalar döndürülür.
2. Her dönen nokta için, sorgu noktası ( $q$ ) ile arasındaki uzaklık hesaplanır ve yakın komşular bildirilir (Andoni ve Indyk, 2006).



Şekil 4.1 Verinin rasgele izdüşümleri alınır. Her izdüşüm birkaç bit ile nicemlenir (Fergus vd., 2008).



Şekil 4.2 Yerel Duyarlı Özetleme algoritmasının gösterimi. Tüm veri tabanındaki imgeler özetlenerek özet tablosu oluşturulur. Aynı özet fonksiyonlarla sorgu imgesi de özetlenir. Benzer imgeler çarpıştır ve çarpışan imgeler içinde yakın komşular aranır (Grauman ve Fergus, 2013).

## 5. YÖNTEM

İmge erişimi için ilk olarak veri tabanının temini, etiketli verilerin temizlenmesi ve imgeleri en iyi şekilde betimleyen öznelik vektörlerinin oluşturulması üzerinde çalışılmıştır.

### 5.1. İmge Veri Tabanının Oluşturulması ve İmgelerin Betimlenmesi

Yapılan tez kapsamında, İspanya'nın Valencia kentindeki "Universitat Politecnica de Valencia" kurumunda çalışan Prof. Roberto Paredes ve ekibinin "Overview of the ImageCLEF 2013 Scalable Concept Image Annotation Task" (Villegas ve Paredes, 2013) projesi kapsamında "Relevant Image Search Engine (RISE) (<http://risenet.iti.upv.es/rise/>)" web sitesi için topladığı 30 milyon civarında imgeden oluşan veri tabanının kullanılması planlanmıştır. Bu doğrultuda ilgili kurumla irtibata geçilerek bu veri tabanına ait yaklaşık 26 milyon imge indirilmiştir. Bu imgelerin 250000 adedi tam otomatik algoritmalar tarafından imge etrafındaki metinler kullanılarak etiketlenmiştir ve bu etiketler arasında oldukça hatalı ve eksik etiketler mevcuttur. İlk olarak birbiriyle iletişim halinde çalışan 3 uzman kişi tarafından bu etiketler üzerinden geçilerek 2,5 ay içerisinde etiketlerdeki hata oranı minimuma indirgenmiş ve 30000 imgeden oluşan bir veri seti oluşturulmuştur. Bu etiketli imge sayısının yeterli olmaması sebebiyle hem etiketleme işlemini hızlandırmak hem de etiketleme iş yükünü azaltmak amacıyla herkesin rahatlıkla kullanabileceği yarı-otomatik bir imge etiketleme aracı Döngel ve Çevikalp (2014) tarafından tasarlanmıştır. Tasarlanan bu yarı-otomatik imge etiketleme aracı genel hatlarıyla aşağıda anlatılmıştır.

#### 5.1.1. Yarı-otomatik imge etiketleme aracı

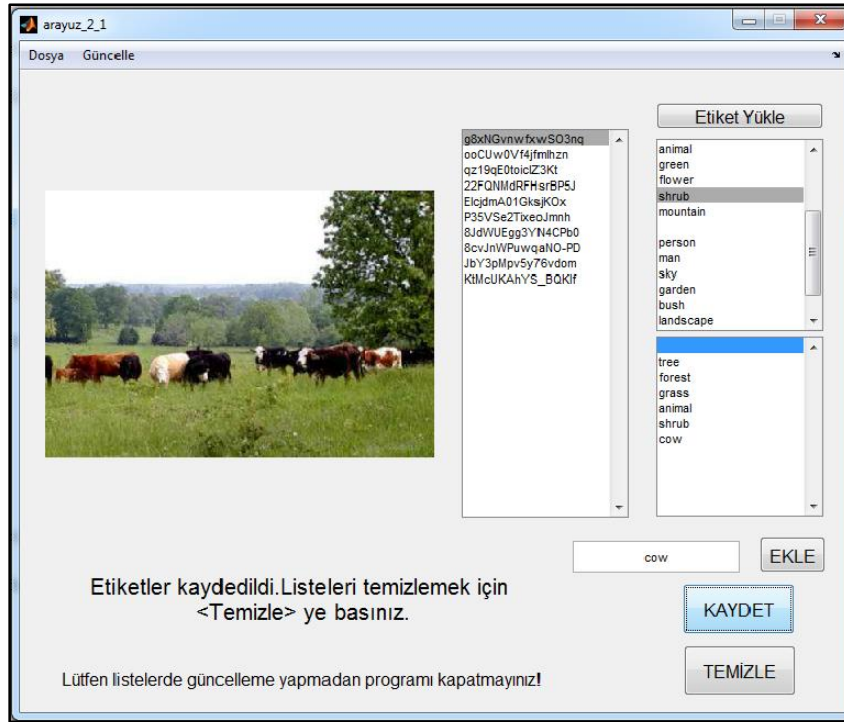
İmge erişimi için ilk başlarda önerilen yöntemlerde düşük seviye öznelik çıkarma ve bu öznelikler arasındaki benzerliklere dayalı teknikler kullanılmıştır. Ancak bu tekniklerle imgeleri doğru bir şekilde sınıflandırmak veya imge içeriğini kolayca ve doğru bir biçimde anlamak pek mümkün değildir. Bu problem anlam bilimsel problem olarak adlandırılır ve bunun çözümünü kolaylaştırmak için yapılacaklardan birisi tüm imgeleri etiketlemektir. Bu şekilde imgeleri sınıflandırmak ve imge içeriğini anlamak daha mümkün

hale gelecektir. Ancak, imge sayısının çok fazla olduğu durumlarda bu işlemi manuel olarak yapmak oldukça zahmetli ve pahalı bir iştir. Bu nedenle bu tür etiketleme işleri yarı-otomatik ve tam-otomatik algoritmalar tarafından gerçekleştirilmektedir. Bu algoritmalara bağlı olarak da çoğunlukla etiketler arasında hatalı ve gereksiz sözcükler bulunmaktadır. Yarı-otomatik algoritmalar kullanılarak oluşturulan hatalı olabilecek etiket bilgileri kullanılmadan önce çözüm getirilmesi gereken önemli ve zorlu bir problem vardır: İçinde hatalı etiket bulunan imgelerin etiketleri olabildiğince düzeltilmeli, yanlış ve gereksiz etiket bilgileri en aza indirgenmelidir. Her bir imgeye ait etiket bilgilerinin insanlar tarafından tek tek kontrol edilip düzeltilmesi pratikte hem zor hem de zaman alıcı bir iştir. Ayrıca etiketleme işinde çalışacak kişilerin elinde belirli bir kategori listesinin bulunmaması nedeni ile farklı kişiler benzer resimleri farklı şekillerde etiketleyebileceğinden imge erişiminde problemlere yol açabilir (Döngel ve Çevikalp, 2014). Bu sorunun çözümü için Döngel ve Çevikalp (2014) tarafından Matlab GUI ortamında yarı-otomatik bir imge etiketleme aracı tasarlanmıştır. Tasarlanan bu aracın genel görünümü Şekil 5.1’de, detaylı görünümleri ise sırasıyla Şekil 5.2, Şekil 5.3 ve Şekil 5.4’de gösterilmektedir. Tasarlanan bu araçla uzman bir kişinin belli bir sayıda etiketleme yapmasıyla kategori listesi oluşturulmuş ve diğer kişilerin bu kategori listesini kullanarak daha pratik ve tutarlı bir şekilde etiketleme yapabilmesi sağlanmıştır. Yarı-otomatik imge etiketleme aracı için Fisher vektör kodları daha oluşturulmadığından kelimeler çantası tabanlı imge betimleme (Csurka vd., 2004; Lazebnik vd., 2006) kullanılmıştır. Kelimeler çantası yönteminde görsel sözlük boyutu 5000 olarak belirlendiğinden her bir imge 5000 boyutlu histogramlarla ifade edilmiştir. Verilen etiketsiz bir sorgu imgesiyle sınırlı sayıdaki etiketli imgelerin 5000 boyutlu öznitelik vektörleri karşılaştırılmıştır. Öznitelikler arasında “Chi-Squared” uzaklık ölçütü kullanılarak en kısa mesafe olanlar belirlenmiştir.  $I_1=(u_1, \dots, u_d)$  ve  $I_2=(v_1, \dots, v_d)$  histogram vektörleri varsa, bu ikisi arasındaki “Chi-Squared” uzaklığı aşağıdaki formül kullanılarak hesaplanır (Döngel ve Çevikalp, 2014):

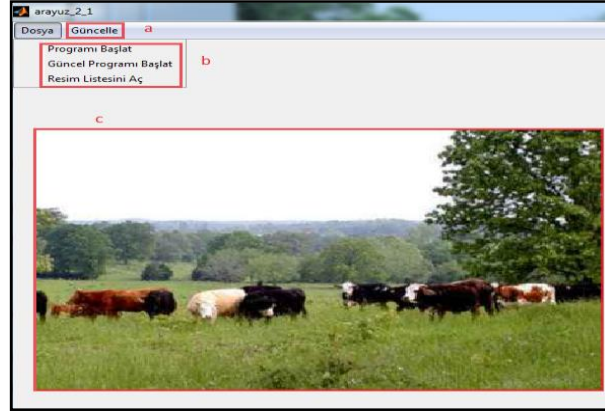
$$CSD(I_1, I_2) = \frac{1}{2} \sum_{m=1}^d \left[ \frac{(u_m - v_m)^2}{(u_m + v_m)} \right] \quad (5.1)$$

Öznitelikler, aralarındaki uzaklık minimum olandan maksimum olana doğru sıralanır ve aralarındaki uzaklık minimum olan 250 öznitelik vektörü ve bu vektörlere karşılık gelen imgeler belirlenir. Belirlenen bu özniteliklere karşılık gelen imgeler ve etiket bilgileri

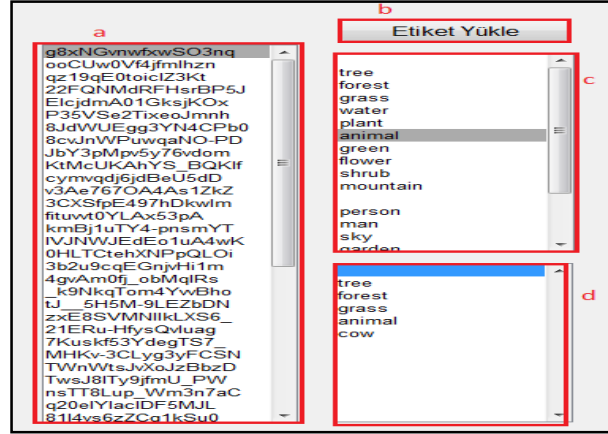
eđitim galerimizde mevcuttur. Bu etiket bilgileri okunarak her bir kategorinin ka defa kullanıldıđı sayılır ve kategoriler kullanım sayılarına gre oktan aza dođru sıralanır. En ok kullanılan 20 kategori belirlenerek etiket listesi oluřturulur. Kullanıcı bu etiket listesinden sorgu imgesiyle alakalı olan etiketleri seer ve sorgu imgesine uygun etiketlerin listelendiđi blme ekler. Eđer bu etiketler dıřında bařka bir etiket eklenmek istenirse o da bu blmdeki etiket listesine eklenebilir. Bu etiket listesi kaydedilerek imge etiketleme iřlemi tamamlanmıř olur. Tasarlanan bu yarı-otomatik imge etiketleme aracıyla hem zaman kaybı hem de insan algısının znelliđi en aza indirgenerek sistematik ve tutarlı etiketleme gerekleřtirilmiř olur (Dngel ve evikalp, 2014).



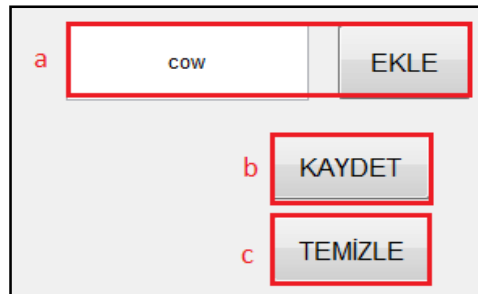
řekil 5.1 Yarı-otomatik imge etiketleme aracının grnm (Dngel ve evikalp, 2014).



Şekil 5.2 (a) “Güncelle” düğmesi. (b) “Programı Başlat”, “Güncel Programı Başlat” ve “Resim Listesini Aç” sekmeleri. (c) Sorgu imgesinin gösterildiği alan. (Döngel ve Çevikalp, 2014)



Şekil 5.3 (a) İmge isim listesinin gösterildiği alan. (b) “Etiket Yükle” düğmesi. (c) Sorgu imgesini etiketlemede kullanılabilecek etiketlerin listelendiği alan. (d) Sorgu imgesine uygun etiketlerin listelendiği alan (Döngel ve Çevikalp, 2014)



Şekil 5.4 (a) Eklenebilecek diğer etiketlerin yazıldığı ve eklendiği alan. (b) Sorgu imgesine uygun etiketlerin kaydedilmesi için gerekli “Kaydet” düğmesi. (c) “Temizle” düğmesi. (Döngel ve Çevikalp, 2014)



Tasarlanan imge etiketleme aracı 300'ü geçen gönüllü öğrenci tarafından uzman kişi gözetimi altında kullanılarak 100700 imge daha kolay, hızlı ve doğru bir şekilde kısa bir süre içinde etiketlenmiştir. Böylece uzman kişilerce manuel olarak etiketlenerek oluşturulan eğitim galerisindeki etiketli imge sayısı, etiketlenen diğer imgeler de eklenerek 30000'den 130700'e yükselmiştir.

### 5.1.2. Fisher vektör ile imgelerin betimlenmesi

Chatfield vd. tarafından (2011) yapılan çalışmada Fisher vektör yönteminin görüntü tanıma kriterlerinde diğer kodlama yöntemlerine göre çok daha iyi sonuçlar verdiği gösterilmiştir. Bu sebeple tez kapsamında imgeleri betimlemek için "Bag of Words" modelleri yerine çok daha iyi sonuçlar veren Fisher vektörlerinin kullanılması planlanmıştır.

Perronin vd. (2010) tarafından geliştirilen Fisher vektör yönteminin kodları bu kişiler tarafından paylaşımına açılmamıştır. Bu yöntemin kodları Oxford Üniversitesi (Chatfield vd., 2011) ve INRIA LEAR takımı tarafından (2012) paylaşımına açılmıştır. Yapılan incelemelerde Oxford Üniversitesince paylaşımına açılan kodlarda ciddi yanlışlıklar olduğu saptanmıştır (Fisher vektör kodlamanın en önemli parçası olan GKM modellerini döndüren algoritmanın çok yanlış sonuçlar döndürdüğü görülmüştür). INRIA LEAR takımı tarafından paylaşımına açılan kodlardaki ufak hatalar düzeltilmiştir. Bu algoritmalara alternatif olarak yeni kodlar oluşturulmuş ve INRIA LEAR takımının kodlarına benzer sonuçlar alınmıştır. Sonuçlara ilişkin detaylı bilgiler deneyler kısmında anlatılacaktır.

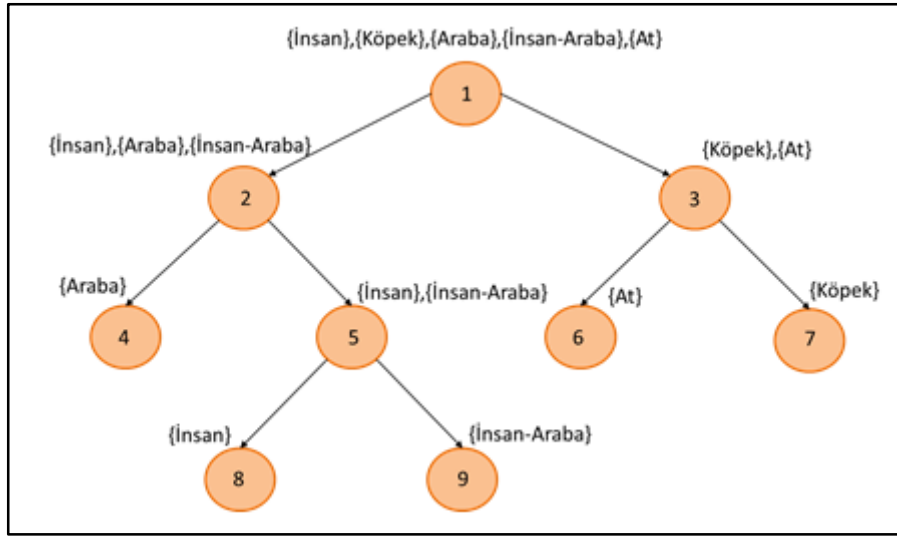
Fisher vektör kodlarının doğruluğunu teyit ettikten sonra Prof. Roberto Paredes ve ekibinin "Overview of the ImageCLEF 2013 Scalable Concept Image Annotation Task" (Villegas ve Paredes, 2013) projesi kapsamında "Relevant Image Search Engine (RISE) (<http://risenet.iti.upv.es/rise/>)" web sitesi için topladığı 30 milyon civarında imgeden oluşan veri tabanındaki imgelerin Fisher vektör kodları elde edilmiştir. Bu işlemler yaklaşık olarak 2 ay sürmüştür.

## 5.2. Sınıf Hiyerarşilerinin Belirlenmesi

Tez kapsamında bir sonraki aşamada etiketli imgeleri kullanarak sınıf hiyerarşilerinin belirlenmesi konusunda çalışmalar yapılmıştır. Bu amaçla ikili sınıf hiyerarşik karar ağaçları kullanılmıştır (binary hierarchical decision trees). Bu yaklaşımda her bir düğümdeki sınıflar iki kümeye ayrılmaktadır. Bu ayırma işlemi sırasında her kümedeki sınıfların elemanları birbirine benzer ve farklı iki kümedeki sınıfların elemanları ise birbirinden olabildiğince farklı ise sınıfların birbirinden ayrılması çok daha kolay olacaktır. Şekil 5.5’de 5 sınıfı ayıran tipik bir ikili sınıf hiyerarşik karar ağacı verilmiştir. Özellikle birbirlerinden uzak olan bu sınıfları en iyi ayıran hiperdüzlemler özet kodlarını üretmek için kullanılırsa başarımlar artacaktır. Bu sebeple karar ağacının her düğümündeki kümeler oluşturulurken iki küme arasındaki boş alan (margin) en büyüklenecek şekilde oluşturulmalıdır.

Rahimi ve Recht’in (2004) çalışmalarında gösterdiği üzere Normalized Cuts (NCuts) toplama algoritması veriyi bölerken bu iki veri arasındaki boş alanı en büyüklenecek bir hiperdüzlem kullanmaktadır. Fakat bu yöntemi direkt olarak sınıf hiyerarşilerini belirlemek için kullanmak iki açıdan sakıncalıdır. Bunlardan ilki NCuts yönteminin eğitimsiz bir yöntem olmasıdır. Bu sebeple de algoritma sonucunda, düğümlerde aynı sınıfa ait olan verilerin aynı gruplarda yer alması, sınıf içi verilerin birbirlerine çok yakın ve diğer sınıflara ait örneklerden çok uzak olmadığı durumlarda garantilenememektedir ve yüksek boyutlu uzaylarda genellikle aynı sınıfa ait veriler birbirinden oldukça uzak olabilmektedir. Bu sorun benzerlik matrisinin oluşturulması sırasında (Wang ve Kumar, 2010; Çevikalp vd., 2008)’de önerildiği gibi aynı sınıfa ait veriler arasındaki benzerlikler en büyük olacak şekilde (en büyük atanabilecek değer 1’dir) seçilerek bir derece azaltılabilmektedir. İkinci problem ise “Spectral Hashing” yönteminde karşılaşılan problem ile aynıdır. Veri sayısının çok fazla olması durumunda benzerlik matrisinin boyutu da büyük olacaktır ve bu matrisi hafızada tutmak mümkün olmamaktadır. Bu sebeple bir yaklaşım kullanma gereği ortaya çıkmaktadır. Bu çalışmada herhangi bir yaklaşım kullanmak yerine (Çevikalp, 2010)’de önerilen yöntem bazı ufak değişikliklerle bu amaç için kullanılmıştır. Bu yaklaşımda NCuts örnek bazında değil sınıf bazında uygulanmaktadır. Bu sayede eğer  $C$  adet sınıf var ise oluşturulacak benzerlik matrisinin boyutu çok daha küçük olan  $C \times C$  olacaktır. Bu yaklaşımı kullanmak için

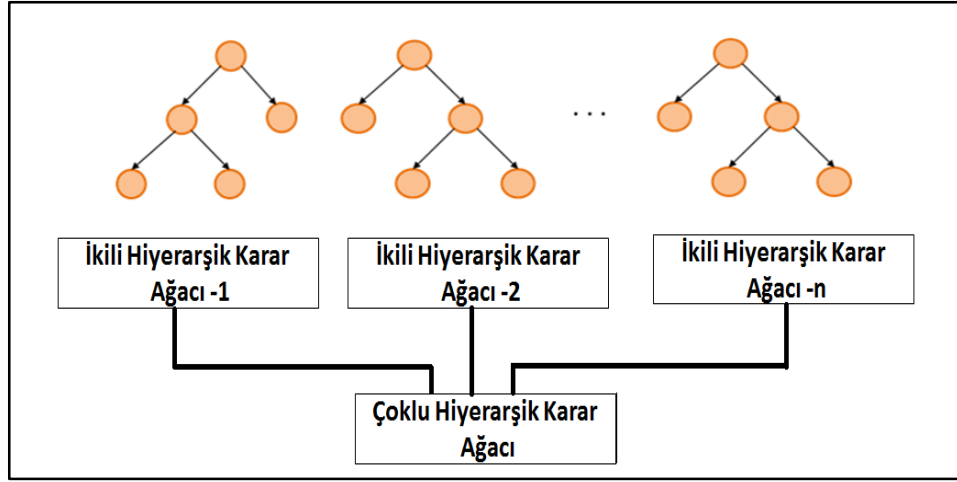
sınıfların modellenmesi ve sınıflar arasında bir benzerlik ölçütü tanımlanması gerekmektedir. Çevikalp'in (2010) önerdiği yöntemde olduğu gibi her bir sınıf, o sınıfa ait verilerin oluşturduğu dışbükey zarf (convex hull) ile modellenmiştir (destek vektör makineleri de aynı yaklaşımı kullanmaktadır). Bu zarflar arasında ikilenik programlama (quadratic programming) kullanılarak elde edilen uzaklıklar ise benzerlikleri oluşturmak için kullanılmıştır.



Şekil 5.5 5 sınıfı ayıran ikili hiyerarşik karar ağacı. Karar ağacı 5'i terminal 4'ü iç düğüm olmak üzere toplam 9 düğüm içermektedir. Her bir iç düğümde sınıflar (1,2,3,6 nolu düğümler) kümeleme yöntemiyle iki gruba bölünmüştür.

Yukarıda anlatılan yöntem her bir veriye ait tek bir etiket olması durumunda rahatlıkla uygulanabilir. Fakat imge erişim uygulamalarında her bir imgeye ait birden fazla etiket olabilir. Mesela bir imgede hem arabalar hem de insanlar varsa bu imge her iki etiketi de ("İnsan" ve "Araba") içerebilir. Bu tür durumları çözmek için birden fazla etiketli gruplar farklı bir kategori olarak ele alınıp daha sonra bu kategori ile ilişkili olduğu sınıflar arasındaki benzerlikler en büyük olacak şekilde seçilerek bir benzerlik matrisi oluşturulabilir. Bu sayede birden fazla sınıflı kategoriler karar ağacının yukarı kısımlarında bağlı oldukları sınıflarla aynı kümelerle düşecek şekilde seçilerek, ayırım daha sağlıklı ayırımın yapılabildiği karar ağacının alt kısımlarına ertelenmiş olur. Bu durum Şekil 5.5'de gösterilmiştir. Bu yöntemle imgeler ve etiket bilgileri kullanılarak sınıflar birbirinden en kolay ayrılacak şekilde bir hiyerarşi oluşturulmaktadır. Önerilen bu yöntemle (Fergus vd., 2010)'de verilen WordNet tabanlı ve imge verilerini kullanmadan sadece kategoriler

arasındaki metin tabanlı anlam bilimsel benzerlikler kullanılarak oluşturulan hiyerarşik yapıdan çok daha üstün ve görsel imge erişimi için daha anlamlıdır. Ayrıca bu çalışmada başarıyı arttırmak için tek bir hiyerarşi oluşturmak yerine birden fazla ağaçtan oluşan bir orman (hierarchical decision forest) kullanılmıştır. Karar ağaçları oluşturulurken rastgele sınıflar ve örnekler seçilerek birden fazla karar ağacı oluşturulmuş ve oluşturulan birden fazla karar ağacı yapısı Şekil 5.6’da gösterilmiştir.



Şekil 5.6 Rastgele sınıflar ve örnekler seçilerek oluşturulan birden fazla karar ağacı örneği

### 5.3. Transdaktif Eğitim

Sınıf hiyerarşileri belirlendikten sonra özet kodları, karar ağacındaki iki kümeyi en iyi ayıran hiperdüzlemi bulan bir ikili sınıflandırıcı kullanılarak oluşturulabilmektedir. Hiperdüzlemin üst tarafında kalan örnekler 1 ve alt tarafında kalan örneklerde 0 olacak şekilde kodlanabilmektedir. Karar ağacının en tepesindeki sınıflandırıcı eğitim kümesindeki tüm imgeleri kullandığından seçilen sınıflandırıcının çok fazla veriyle sorunsuz bir şekilde çalışabilen bir sınıflandırıcı olması gerekmektedir. Şu an literatürde milyonlarca veriyle çalışabilecek sınıflandırıcılar büyük boşluk yaklaşımını kullanan destek vektör makineleri (Cortes ve Vapnik, 1995) ve türevleriyle, Çevikalp ve arkadaşlarının (2010) çalışması aynı yaklaşımı kullanan sınıflandırıcıdır. Bu önerilen yöntemde sınıflar dışbükey zarflar yerine affine alt uzaylar kullanarak modellenmiş ve iki affine alt uzayını aralarında maksimum boşluk kalacak şekilde ayıran bir hiperdüzlem bulunmuştur. Bu yöntemlerin ortak noktaları iki sınıfı en iyi şekilde ayıran hiperdüzlemi bulma probleminin ikilenik programlama olarak formüle edilmesidir. İkilenik programda

Sequential Minimal Optimization (SMO) algoritması (Platt, 1999) kullanılarak çözülmektedir. SMO tekniği veriyle aynı boyuttaki Hessian matrisini kullanmak yerine iteratif olarak sadece kısıtları bozan iki örneği kullanmaktadır. Bu sayede hafızaya sığdırılabilen tüm veri üzerinde bu sınıflandırıcılar rahatlıkla çalıştırılabilmektedir.

İmge erişimi probleminde genellikle etiketli verilerin yanında çok sayıda da etiketsiz veri bulunmaktadır. Eğitim kümesindeki etiketli veriler kümesinin ( $L = \{(x_1, y_1), \dots, (x_l, y_l)\}$ )  $l$  örnekten oluştuğunu ve buna ek olarak etiketsiz veriler kümesinin de ( $U = \{(x_{l+1}), \dots, (x_{l+u})\}$ )  $u$  adet örnekten oluştuğunu varsayalım. Yukarıda bahsedilen geleneksel sınıflandırıcılar eğitimli yaklaşımlar olup sadece etiketli veriyi kullanabilmektedir. Eğitilen sınıflandırıcılar daha sonra  $Y = (y_{l+1}, \dots, y_{l+u})$  etiketlerinin değerini tahmin etmek için kullanılmaktadır. Geleneksel eğitimli öğrenme ile eğitilen sınıflandırıcılarda eğer örnek sayısı,  $l$ , çok az ve buna karşın sınaama yapılan örnek sayısı,  $u$ , çok fazla ise sınaama kümesinde bulunan örneklere ait öznitelik verilerinin öğrenme süreci içerisinde kullanılması ve böylelikle eldeki tüm veriler kullanılarak öznitelik verilerine ait ( $X$ ) dağılımının daha gerçekçi kestirimidir. Diğer bir söyleyişle transdaktif öğrenme ile sadece etiketli veriler yerine hem etiketli hem de etiketsiz veriler ( $X = (x_1, \dots, x_l, x_{l+1}, \dots, x_{l+u})$ ) kullanılarak, çok daha başarılı bir sınıflandırıcı elde edilebilecektir.

Transdaktif öğrenme ilk defa Vapnik tarafından ortaya atılmışsa bile bu işlevi yerine getiren ilk algoritma Bennett ve Demiriz (1998) tarafından önerilmiştir. Daha sonra bu konuda çok sayıda yayın yapılmıştır (Joachims, 1999; Collobert vd., 2006). Transdaktif öğrenmede maliyet fonksiyonu konveks olmadığından çözümünü oldukça zordur ve Collobert ve diğerleri tarafından (2006) önerilen yöntem kadar, önerilen yöntemler sadece veri sayısı çok az olan problemlerde kullanılabilmiştir. Collobert vd. (2006) ise son yıllarda önerilen “concave-convex procedure” tekniğini kullanarak bu yöntemi veri sayısının çok fazla olduğu durumlarda da kullanılabilir bir yöntem haline getirmişlerdir.

Transdaktif destek vektör makinelerinde  $l$  örnekten oluşan “+” ve “-” olarak adlandırılan iki sınıfa ait etiketli veriler ( $L = \{(x_1, y_1), \dots, (x_l, y_l)\}$ ,  $y \in \{1, -1\}$ ) ile  $u$  adet etiketsiz veri ( $U = \{(x_{l+1}), \dots, (x_{l+u})\}$ ) vardır. Burada amaç etiketli ve etiketsiz veriyi aralarında maksimum boşluk (margin) olacak şekilde ayırmaktır. Şekil 5.7’de görüldüğü

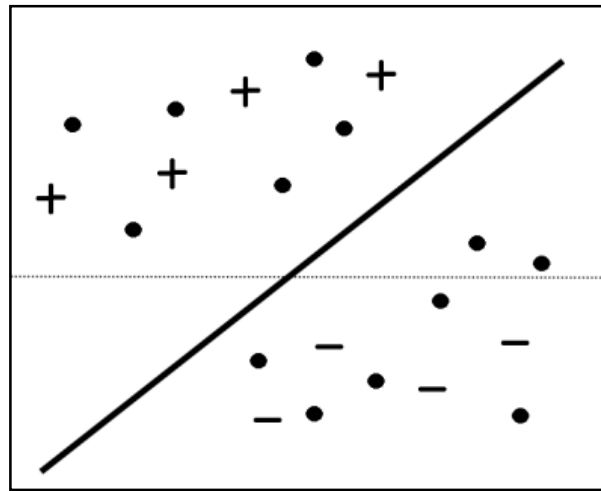
gibi etiketsiz veri de işin içine katıldığında çok daha iyi bir ayırıcı hiperdüzlem bulunabilmektedir. Transdaktif destek vektör makinelerinde etiketli veriler üzerinde destek vektör makinelerinde kullanılan klasik Hinge maliyet fonksiyonu uygulanmaktadır. Etiketsiz verinin de iki sınıf arasındaki boşluktan uzakta kalması istenmektedir. Bu durumda verileri en iyi ayıran hiperdüzlemi bulma problemi aşağıdaki gibi yazılabilmektedir:

$$\min_{w,b,\xi \geq 0} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i + C^* \sum_{i=1}^{l+u} \xi_i \quad (5.2)$$

$$\text{subject to } y_i \langle w, x_i + b \rangle \geq 1 - \xi_i, \quad i = 1, \dots, l$$

$$|\langle w, x_i \rangle + b| \geq 1 - \xi_i, \quad i = 1, \dots, l + u$$

Denklem 5.2’de  $w$ , ayırıcı hiperdüzlemin normalini,  $b$  ise hiperdüzlemin ofset parametresini göstermektedir. Hiperdüzlemin üstünde kalan örnekler “+” sınıfına, altında kalanlar ise “-” sınıfına atanacaktır. Denklem 5.2’de verilen problem konveks bir problem değildir ve çözümü de zordur. Collobert ve diğerleri (2006) bu problemi “concave-convex procedure” tekniği kullanarak her bir iterasyonda konveks bir ikilenik programı çözen bir yönteme indirgeyerek, transdaktif öğrenme yöntemini çok fazla veri ile kullanmanın önünü açmışlardır. Bizde bu tez çalışmasında Collobert ve diğerleri (2006) tarafından geliştirilen bu yöntemi kullandık.



Şekil 5.7 Transdaktif Destek Vektör Makineleri ile öğrenme. Sadece etiketli veri kullanıldığında destek vektör makineleri veriyi ayırmak için kesik çizgi ile gösterilen hiperdüzlemi döndürmektedir. Fakat etiketsiz veri (dairelerle gösterilen veriler) de işin

içine katıldığında koyu renk ile belirtilen çok daha iyi bir ayırıcı hiperdüzlem bulunmaktadır.

Transdaktif öğrenmenin geleneksel sınıflandırma yöntemlerine göre avantajı, az sayıda olan öğrenme verilerinin bulunmasına karşın çok sayıda ve kolayca elde edilebilen etiketsiz test veri kümelerinin bulunduğu durumlarda görülebilmektedir.

Yapılan çalışmada rasgele sınıflar ve örnekler seçilerek birden fazla hiyerarşik karar ağacı oluşturulmuş ve ardından sınıf hiyerarşileri belirlenmiştir. Elde edilen karar ağaçlarının düğüm noktalarına transdaktif destek vektör makineleri uygulanarak farklı destek vektör makine sınıflandırıcısı modelleri elde edilmiştir. Destek vektör makine sınıflandırıcısının döndürdüğü hiperdüzlemlerin normal vektörleri kullanılarak veri boyutu indirgenmiş ya da bunlar doğrudan özetleme amaçlı olarak kullanılmıştır.

#### 5.4. Özetleme Yöntemi

Sınıf hiyerarşileri belirlendikten sonra karar ağacının düğüm noktalarına transdaktif destek vektör makineleri uygulanmak suretiyle elde edilen modellerle yeni bir imge geldiğinde o imgenin sadece kategorisi belirlenebilmektedir. Oysa imge erişiminde amaç sınıflandırma değil, veri tabanında sorgu imgesine benzer imgeleri döndürmektir. Bu sebeple sorgu imgesinin kategorisi belli olduktan sonra o kategori (bu kategoriye yakın kategoriler de dahil edilebilir) içerisinde sorgu imgesine benzer imgelerin bulunması gereklidir. Aynı kategorideki tüm imgeleri benzer imgeler olarak sıralamak sakıncalıdır. Çünkü aynı kategori altında mısır örneğinde olduğu gibi (Arama motoruna mısır yazıldığında arama motoru hem sebze olan mısır resimleri hem de Mısır'a ait fotoğraflar döndürebilmektedir) birbirinden farklı imgeler olabilmektedir. Ya da örneğin bir araba verilmişse amaç, bu arabaya en benzer araba resimlerini içeren imgeleri bulmak olabilmektedir. Aynı kategori içerisinde sorgu imgesine en çok benzeyen imgelerin döndürülmesi işlemini de hızlı ve etkili bir şekilde yapmak için Özetleme yöntemi kullanılmıştır.

Özetleme yönteminde her bir imge  $b$ -bitten oluşan ve özet vektörü adı verilen bir vektör ile ifade edilmektedir. Özet vektörleri hafızada daha az yer tutacak bir şekilde saklanmaktadır. Ayrıca bit düzeyinde işlemler CPU tarafından çok hızlı bir şekilde

gerçekleştirildiğinden, özetleme tabanlı imge betimleme kullanılarak sorgu imgesine benzer imgeleri bulma işlemi oldukça hızlanmaktadır. Özetleme işleminde amaç  $k$  boyutlu örnek uzayında imge histogramlarının arasındaki uzaklıkları koruyan özet kodları üretmektir. Bu nedenle yöntem genellikle Yerel Duyarlı Özetleme olarak adlandırılmaktadır.

Bu çalışmada Andoni ve Indky (2004) tarafından önerilen “Exact Euclidean Locality Sensitive Hashing” (E2LSH) yöntemi kullanılmıştır. E2LSH yöntemi, Datar ve diğerlerinin (2004) önerdiği Yerel Duyarlı Özetleme yöntemine dayanmaktadır. Bu algoritmanın performansını etkileyen üç ana parametre bulunmaktadır: her bir özet değerinin izdüşüm sayısı ( $k$ ), özet tablosu sayısı ( $L$ ) ve izdüşüm genişliği ( $w$ ).  $k$  parametresi özet değerlerini hesaplamada harcanan zaman ve hatalı pozitifleri ayıklamada harcanan zaman arasında ikilem oluşturmaktadır. Örneğin, sorgu noktası ve diğer noktalar arasındaki uzaklık hesaplamada, büyük  $k$  değeri özet hesaplama sayısını artırmaktadır. İzdüşüm genişliğini ( $w$ ) azaltma ise herhangi iki noktanın çarpışma olasılığını azaltmaktadır.

Yapılan çalışmada birden fazla karar ağacı kullanılarak bu karar ağaçlarının düğüm noktalarına transdaktif destek vektör makineleri uygulanmıştır. Bunun sonucunda farklı DVM sınıflandırıcısı modelleri elde edilmiştir. DVM sınıflandırıcısının döndürdüğü hiperdüzlemlerin normal vektörleri kullanılarak veri boyutu indirgenmiştir. Bu aşamadan sonra sıra boyutu indirgenmiş verinin özet değerlerini bulmaya gelmiştir. Bu amaç için Shakhnarovich (2009) tarafından Matlab aracı kullanılarak kodlanan E2LSH yöntemi kullanılmıştır. Burada ilk olarak kendi belirlediğimiz her bir özet değerinin izdüşüm sayısı ( $k$ ), özet tablosu sayısı ( $L$ ) ve izdüşüm genişliği ( $w$ ) verilerek *lsh* komutuyla verinin özet tablosu oluşturulmuştur. Diğer bir deyişle tekrarlı olarak farklı fonksiyonlarla özet işleminin yapılmasının ardından veri, özet tablosunda kovaların içine konulmuştur.

Yapılan özet işleminin tekrarlı olarak farklı fonksiyonlarla yapılmasıyla, tekrarlar sonucunda aynı özet kovasına en az bir kere girenlerin benzer olabileceğine, girmeyenlerin ise benzemedikleri hükmüne varılabilmektedir. Bu arada bu yöntemle hata yapma olasılığı da vardır. Yanlış pozitif hatalar, aynı kovaya girip de benzemeyenler, gereksiz yere işlem



yüküne sebep olmaktadır. Doğru negatifler ise aynı kovaya girmemelerine rağmen benzer olanlardır.

Özet tablosundaki kova sayısının arama süresi üzerinde etkisi büyüktür. Kova sayısı, özet değerinin izdüşüm sayısı ( $k$ ), özet tablosu sayısı ( $L$ ), izdüşüm genişliğine ( $w$ ) ve veri miktarına bağlı olarak değişiklik göstermektedir. Veriye bağlı olarak kovaların çoğunda sadece birkaç örnek bulunması durumu (örneğin 50000 örnekli bir veri 10000 tane kovaya konulmuş ise her bir kovaya az sayıda örnek düşmektedir), kova içine örnek dağılımının makul olduğu anlamına gelir ve bu sayede kovalar içinde arama işlemi daha hızlı ve kolay olur. Kova sayısının çok az olduğu durumlardan kaçınmamız gerekmektedir. Çünkü bu durumda bazı kovaların içerisindeki örnek sayısı oldukça fazladır, bu da kova içerisindeki arama süresini uzatacağından performansı olumsuz etkileyecektir. Böyle bir durum söz konusu olduğunda özet değerinin izdüşüm sayısı ( $k$ ) az bir oranda artırılabilir.

Özet tablosu oluşturma işleminden sonra sıra sorgulama işlemine gelmiştir. Öncelikle verilen sorgu imgesine karşılık gelen özet kodu bulunup ardından özet tablosundaki kovalar arasında bu özet kod karşılaştırma yoluyla aranmaktadır. Özet tablosundan sorgu imgesine ait özet kodun aranması işlemi *lshlookup* komutuyla yapılmaktadır. Bu komuta eğitim verisi, sorgu imgesi, özet tablosu ve sorgu imgesine en yakın komşuların sayısı girdi olarak verilmektedir. Bu sayede komut, veri tabanında sorgu imgesine benzer imgeleri belirtilen en yakın komşu sayısına göre döndürmektedir. Başka bir deyişle özet kodlar oluşturulduktan sonra, yeni bir imge geldiğinde özet kodu oluşturularak, veri tabanındaki imgelerin özet kodları ile karşılaştırma yapılarak çok az bir sürede veri tabanından bu sorgu imgesine benzeyen imgeler döndürülmektedir.

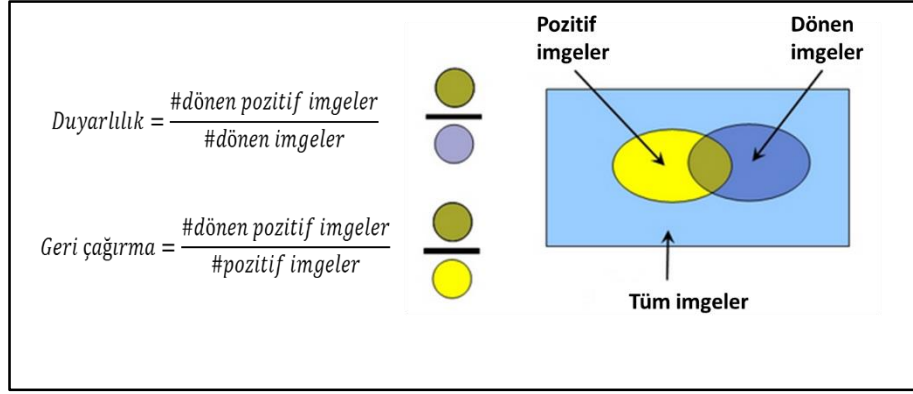
## 6. BULGULAR VE TARTIŞMA

Deneysel çalışmalar farklı etiketli veri tabanları üzerinde gerçekleştirilmiştir: Pascal VOC 2007, Caltech 256, AR, UC Merced Land Use, CIFAR-10, INRIA Holidays ve Kentucky veri tabanı. Ayrıca büyük veri tabanlarında yapılan imge erişimlerinde Flickr1M veri tabanı da bu veri tabanlarına ek olarak kullanılmıştır. Deneysel çalışmalara ilk olarak imgelerin Fisher vektör ile betimlenmesinden başlanılmıştır. Ardından sırasıyla sınıf hiyerarşilerinin belirlenmesi, transdaktif öğrenme ve yerel duyarlı özetleme konularında deneyler yapılmıştır. Bu kısımda yapılan deneyler ve sonuçları detaylı bir şekilde anlatılacaktır.

İmge erişim sisteminin performansını değerlendirmek amacıyla çeşitli ölçütler vardır ve bunlardan en yaygın olanı mAP (mean average precision)'dir. Her sorgu için AP hesaplamak için, öncelikle iki ölçütün duyarlılık ve geri çağırmanın hesaplanması gerekmektedir. Duyarlılık (precision,  $p$ ), dönen pozitif imgelerin toplam dönen imge sayısına oranı olarak ifade edilir. Geri çağırma (recall,  $r$ ) ise dönen pozitif imgelerin toplam pozitif imge sayısına oranı olarak ifade edilir (Şekil 6.1). Duyarlılık ve geri çağırma değerleri hesaplanarak duyarlılık / geri çağırma eğrisi  $p(r)$  çizilir ve ardından bu eğrinin altında kalan alan hesaplanarak denklem 6.1'de ifade edilen ortalama duyarlılık (average precision, AP) değeri elde edilir. Denklem 6.1'deki  $P(k)$  ve  $r(k)$  sırasıyla duyarlılık ve geri çağırma değerlerini,  $n$  ise dönen imge sayısını göstermektedir. Sorgu kümesi için mAP değeri her sorgu için AP değerinin ortalaması alınarak denklem 6.2 ile hesaplanır. Denklemden  $Q$  sorgu sayısını göstermektedir.

$$AP = \int_0^1 p(r) dr \approx \sum_{k=1}^n P(k) \Delta r(k) \quad (6.1)$$

$$mAP = \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (6.2)$$



Şekil 6.1 Duyarlılık ve geri çağırma gösterimi

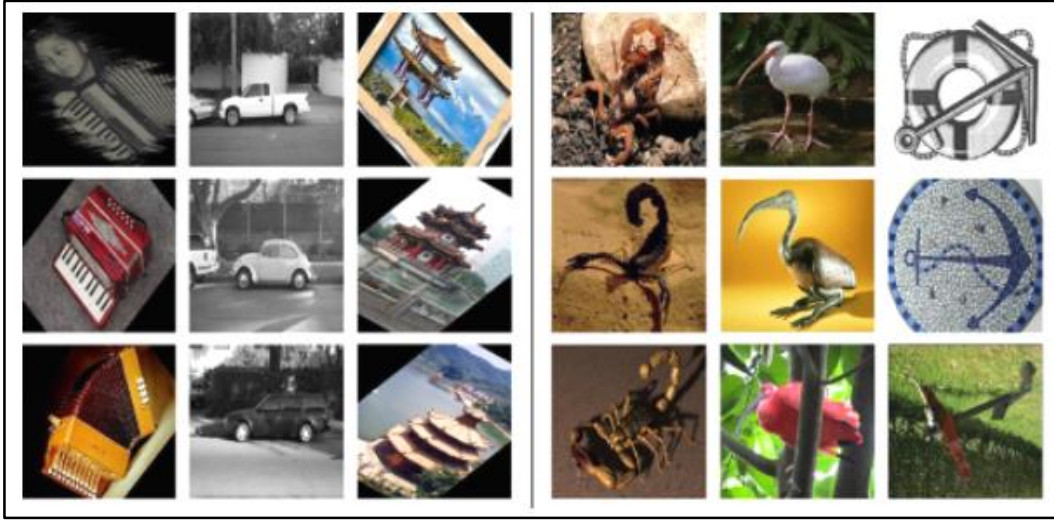
(<http://www.di.ens.fr/willow/events/cvml2011/materials/practical-classification/>)

Deneysel performans ölçütü olarak yukarıda bahsedilen duyarlılık/geri çağırma eğrisinin altında kalan alan olarak hesaplanan ortalama duyarlılık imge erişimi için kullanılmıştır. mAP tüm kategoriler üzerinde AP ortalamasını ölçmektedir. Her bir sorgu imgesi için duyarlılık/geri çağırma eğrisi elde edilmiş ve onun AP'ı hesaplanmıştır. Sonrasında da tüm sorgu kümesinin ortalaması alınmıştır.

### 6.1. Fisher Vektör ile İmgelerin Betimlenmesine Dair Deney

Inria ve Oxford Üniversitesi tarafından paylaşılan Fisher vektör kodlarına alternatif olarak yeni kodlar oluşturulmuştur. Fisher vektör kodlarının doğruluğunu teyit etmek amacıyla PASCAL VOC 2007 (Everingham vd., 2010) veri tabanı üzerinde deneyler yapılmıştır. PASCAL VOC 2007 veri tabanı 20 sınıfa ait etiketli imgelerden oluşmaktadır. Şekil 6.2'de bu veri tabanına ait örnek imgeler gösterilmiştir.

PASCAL VOC 2007 veri tabanı üzerinde Fisher vektör imge betimlemesi kullanılarak Doğrusal Destek Vektör Makineleri sınıflandırıcısı ile Çizelge 6.1'de verilen sonuçlar alınmıştır. Fisher vektörler için 256 Gauss karışım modeli kullanılmış olup, yamaları betimlemek için SIFT özneliklerinin yanı sıra literatürdeki LBP, LTP, SURF, FT (Fourier Transform) gibi diğer önemli öznelikler ve onların kombinasyonları kullanılmıştır. Performans ölçütü olarak mAP (mean average precision) kullanılmıştır. Çizelge 6.1'de görüldüğü üzere sadece doğrusal destek vektör makineleri kullanılarak, PASCAL VOC 2007 yarışmasında elde edilen en iyi sınıflandırma sonuçlarına benzer sonuçlar elde edilmiştir. Bu da kodların düzgün bir şekilde çalıştığını göstermektedir.



Şekil 6.2 PASCAL VOC 2007 veri tabanına ait çeşitli örnek imgeler.

Çizelge 6.1 PASCAL VOC veri tabanı üzerinde Fisher vektör betimlemesi kullanılarak elde edilen tanıma oranları.

Class \ Method	VOC 2007 Winner	FT	SIFT	SURF	LBP	LTP	FT+SIFT	FT+SURF	FT+LBP	FT+LTP	FT+SIFT+LTP
Aeroplane	77.5	73.2	77.6	60.9	69.6	68.1	77.8	70.1	75.8	72.0	<b>78.2</b>
Bicycle	63.6	49.1	65.2	39.7	45.4	50.0	65.2	51.5	53.9	52.6	<b>66.3</b>
Bird	<b>56.1</b>	39.1	51.7	32.8	39.5	37.6	53.8	46.2	45.1	44.0	55.1
Boat	<b>71.9</b>	61.7	67.4	37.9	54.4	63.4	69.6	55.3	66.5	64.5	71.1
Bottle	<b>33.1</b>	22.9	29.9	10.9	22.6	21.4	29.6	21.3	20.6	24.5	29.5
Bus	60.6	48.0	65.3	33.0	47.2	46.5	65.0	49.5	53.5	52.7	<b>66.4</b>
Car	78.0	67.6	77.2	60.7	66.7	68.1	77.1	70.0	71.6	71.1	<b>78.1</b>
Cat	<b>58.8</b>	43.5	57.0	29.1	43.7	47.2	55.6	44.7	49.7	51.2	57.3
Chair	<b>53.5</b>	45.9	51.1	34.8	42.9	45.0	50.8	42.4	47.1	46.5	52.0
Cow	42.6	33.8	<b>45.0</b>	20.9	26.3	36.1	45.3	26.9	32.9	36.9	44.6
Diningtable	54.9	42.7	53.5	27.3	38.9	43.6	54.4	39.7	44.0	45.0	<b>56.7</b>
Dog	<b>45.8</b>	35.9	40.4	19.8	35.7	32.8	40.2	29.8	38.8	37.5	42.3
Horse	77.5	69.1	78.6	62.1	70.6	69.2	<b>78.9</b>	70.5	73.0	72.5	78.6
Motorbike	64.0	50.4	67.7	43.5	55.1	49.1	<b>68.6</b>	56.1	58.5	56.5	<b>68.6</b>
Person	<b>85.9</b>	77.7	82.0	67.7	75.4	76.2	83.2	75.8	79.9	78.8	83.3
Pottedplant	<b>36.3</b>	22.5	28.6	11.4	21.8	24.3	28.3	20.8	24.8	24.9	28.5
Sheep	44.7	35.2	47.6	18.7	26.5	33.9	46.7	31.9	35.0	35.9	<b>47.9</b>
Sofa	50.9	41.1	49.4	37.4	38.4	38.6	50.6	44.4	48.7	43.2	<b>51.5</b>
Train	79.2	69.8	78.7	47.8	66.6	71.9	78.6	68.8	74.6	72.1	<b>80.2</b>
Tvmonitor	<b>53.2</b>	35.8	51.9	29.7	38.9	37.9	50.8	38.8	41.9	39.1	51.8
Average	<b>59.4</b>	48.3	58.3	36.3	46.3	48.0	58.5	47.4	51.8	51.1	<b>59.4</b>

## 6.2. Sınıf Hiyerarşilerinin Belirlenmesine Dair Deneyler

Etiketli imgeleri kullanarak sınıf hiyerarşilerinin belirlenmesi konusunda çalışmalar yapılmış ve bu amaçla ikili sınıf hiyerarşik karar ağaçları kullanılmıştır (binary hierarchical decision trees). Bu yöntemin başarımı ilk olarak büyük ölçekli veri tabanı olarak kabul edilen Caltech 256 (Griffin, 2007) veri tabanı üzerinde test edilmiştir. Bu veri tabanındaki her imge için tek bir etiket bilgisi olduğundan (Çevikalp, 2010)'de önerilen yöntem hiç bir değişiklik yapılmadan uygulanabilmektedir. Bu veri tabanındaki başarımı test etmek için literatürde kullanılan standart yaklaşım kullanılmıştır: Her bir sınıftan rastgele 30 örnek eğitim kümesi için, diğer bir 30 örnek ise test kümesi için kullanılmıştır. Eğitim kümesindeki veriler kullanılarak Fisher vektör parametreleri elde edilmiş, daha sonra bunlar hem eğitim hem de test kümesindeki imgelerin Fisher vektörlerini oluşturmak için kullanılmıştır. Her bir düğümdeki örnekler doğrusal Destek Vektör Makineleri kullanılarak ayrılmıştır. Bu yöntemin başarısını en-yakın komşu ve bire-karşı-tüm yaklaşımını kullanan Destek Vektör Makineleri sınıflandırıcıları ile karşılaştırılmıştır. Sonuçlar Çizelge 6.2'de verilmiştir. Önerilen yöntem en-yakın komşu sınıflandırıcısına oranla çok daha iyi sonuçlar verirken bire-karşı-tüm yaklaşımını kullanan Destek Vektör Makinelerine oranla çok daha kötü sonuçlar vermiştir. Bunun üzerine başarımı arttırmak için tek bir hiyerarşi oluşturmak yerine birden fazla hiyerarşi oluşturulmuştur. Bunu gerçekleştirmek için her bir eğitim sınıfından 30 örnek yerine rassal olarak 20 örnek seçilmiştir. Her bir hiyerarşi ağacı farklı bir etiket döndürebileceğinden nihai sınıf ataması kararı için en fazla oyu alan sınıf seçilmiştir. Bu yaklaşımın sınıflandırma başarısını önemli ölçüde arttırdığı gözlenmiştir. Bu sebeple bu çalışmada tek bir hiyerarşi ağacı yerine birden fazla ağaçtan oluşan bir orman (hierarchical decision forest) kullanılmasına karar verilmiştir.

Çizelge 6.2 Caltech 256 veri tabanı üzerinde elde edilen tanıma oranları (%).

Sınıflandırıcılar	Sınıflandırma Başarımı
k-en yakın komşu	10.66
Bire-karşı-tüm DVM	36.30
1 karar ağacı	20.07
7 karar ağacı	27.63
14 karar ağacı	30.52
21 karar ağacı	30.46
28 karar ağacı	31.30
35 karar ağacı	32.03
42 karar ağacı	32.36

### 6.3. Transdaktif Öğrenmeye Dair Deneyler

Sınıf hiyerarşilerinin belirlenmesi aşamasından sonra özet kodlarının oluşturulabilmesi için karar ağacındaki iki kümeyi en iyi ayıran hiperdüzlemi bulan bir ikili sınıflandırıcı kullanılmaktadır. Karar ağacının en tepesindeki sınıflandırıcı eğitim kümesindeki tüm imgeleri kullandığından seçilen sınıflandırıcının çok fazla veriyle sorunsuz bir şekilde çalışabilen bir sınıflandırıcı olması gerekmektedir. Bu sebeple bu tez kapsamında sınıflandırıcı olarak transdaktif destek vektör makine yöntemi kullanılmış ve yönteminin sınıflandırma başarısını geliştirmek üzere Caltech-256 ve AR yüz veri tabanları üzerinde çalışmalar yapılmıştır. Yapılan bu çalışmalar detaylı bir şekilde Döngel ve Çevikalp'in (2015) çalışmasında anlatılmıştır. Kısaca yapılanlardan ve deneysel sonuçlardan bahsetmek gerekirse, öncelikle transdaktif öğrenme ve ikili sınıf hiyerarşik karar ağaçları yöntemleri birlikte kullanılarak ayırt-edici izdüşüm vektörleri bulunmuştur. Ardından bu izdüşüm vektörleri verilerin boyutlarını indirgemedi kullanılarak k-en yakın komşu sınıflandırıcısının performansı önemli ölçüde geliştirilmiştir. Sınıf hiyerarşilerinin belirlenmesine dair yapılan deneylerden elde edilen sonuçlara göre tek bir hiyerarşi ağacı yerine birden fazla ağaçtan oluşan bir orman kullanılmasının daha iyi sonuçlar verdiği gözlemlenmiştir. Bu sebeple de yapılan bu çalışmada karar ağaçları oluşturulurken rastgele sınıflar ve bu sınıflara ait rastgele örnekler seçilmek suretiyle birden fazla karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırt etmek için de transdaktif destek vektör makineleri kullanılmıştır. Destek vektör makinelerinin

döndürdüğü hiperdüzlemlerin normal vektörleri,  $w$ , izdüşüm vektörleri olarak seçilerek veri boyutunu indirgemek için kullanılmıştır. Boyutu indirgenmiş eğitim ve test verisi arasındaki uzaklıklar Öklid uzaklık ölçütü kullanılarak ayrı ayrı hesaplanmıştır. Bu şekilde test verisine ait farklı tahmini etiket bilgileri elde edilmiştir. Bu etiket bilgilerine çoğunluk oylama (majority voting) uygulanarak test verisine ait son etiket belirlenmiştir. Bunun yanında hesaplanan eğitim ve sınav verileri arasındaki uzaklıkların sırasıyla ortalaması, minimumu, çarpımı ve medyanı hesaplanmıştır. Hesaplanan her bir yeni uzaklığın minimumu bulunarak test verisindeki etiketler tahmin edilmiş ve sınıflandırma başarı oranları hesaplanmıştır.

Ayrıca veri boyutunu indirgemek için kullanılan DVM sınıflandırıcısının döndürdüğü hiperdüzlemlerin normal vektörlerine,  $w$ , 2 farklı işlem uygulanarak yukarıda anlatılan sınav işlemleri tekrar edilmiştir. Bu işlemlerden birincisinde  $w$ , 'nin normu 1'e eşitlenerek elde edilen vektörler kullanılarak izdüşüm matrisi  $w_2$ , ikincisinde ise  $w$  vektör setlerinin ortonormali alınarak ortonormal vektörlerden oluşan  $w_3$  bulunmuştur. Bulunan bu izdüşüm matrisleri kullanılarak da boyut indirgemesi yapılmış ve sınav işlemleri tekrarlanmıştır.

Önerilen yöntemin sınıflandırma performansını değerlendirmek için yapılan deneylerde Caltech-256 ve AR yüz veri tabanları kullanılmıştır. Caltech-256 veri tabanı üzerinde yapılan deneylerin sonuçları Çizelge 6.3'de gösterilmiştir. Çizelge 6.3 ve 6.4'te  $w_1$ , hiperdüzlemlerin normallerine hiç bir değişiklik yapılmadan kullanılan izdüşüm vektörlerini göstermektedir. Bu deneyde en yüksek sonuç olan %25,59 uzaklıkların ortalaması alınarak elde edilmiştir ve bu sonuç, en yakın komşu sınıflandırıcısı ile elde edilen %10,66'lık sonuca oranla çok daha iyidir. AR yüz veri tabanı üzerinde yapılan deneylerin sonuçları Çizelge 6.4'te gösterilmiştir. Elde edilen sonuçlara göre önerdiğimiz yöntem kullanılarak %99'luk bir sınıflandırma başarısı elde edilmiştir. En yakın komşu sınıflandırıcısı ile elde edilen sonuç ise %74,96 dır (Döngel ve Çevikalp, 2015).

Caltech-256 ve AR veri tabanları üzerinde yapılan deneylerden elde edilen sonuçlara göre; transdaktif destek vektör makineleri ile ikili karar ağaçları yöntemi birlikte kullanılarak boyut indirgeme için kullanılması en yakın komşu sınıflandırma başarısını

büyük ölçüde artırdığı gözlemlenmiştir. Bunun yanı sıra boyut da azaltıldığından sınıflandırma için geçecek zamanda önemli kazanımlar elde edildiği gözlemlenmiştir (Döngel ve Çevikalp, 2015).

Çizelge 6.3 Caltech -256 üzerindeki deney sonuçları (Döngel ve Çevikalp, 2015)

Yöntem		Sınıflandırma Başarımları (%)		
		$w_1$	$w_2$	$w_3$
Önerilen yöntem	Çoğunluk oylama	25,53	24,94	23,60
	Ortalama	<b>25,59</b>	25,53	23,98
	Minimum	19,96	19,54	18,95
	Çarpım	25,29	24,97	23,67
	Medyan	24,14	23,64	22,43
En yakın komşu sınıflandırıcısı		10,66		

Çizelge 6.4 AR veri tabanı üzerindeki deney sonuçları (Döngel ve Çevikalp, 2015)

Yöntem		Sınıflandırma Başarımları (%)		
		$w_1$	$w_2$	$w_3$
Önerilen yöntem	Çoğunluk oylama	98,80 , $\sigma=0,95$	98,53 , $\sigma=0,86$	98,57 , $\sigma=0,74$
	Ortalama	98,93 , $\sigma=0,91$	97,90 , $\sigma=1,68$	98,20 , $\sigma=1,34$
	Minimum	97,80 , $\sigma=1,28$	97,43 , $\sigma=0,99$	97,43 , $\sigma=1,12$
	Çarpım	98,90 , $\sigma=0,75$	98,03 , $\sigma=1,37$	98,37 , $\sigma=1,01$
	Medyan	<b>99</b> , $\sigma=0,82$	98,13 , $\sigma=1,25$	98,43 , $\sigma=1,04$
En yakın komşu sınıflandırıcısı		74,96		



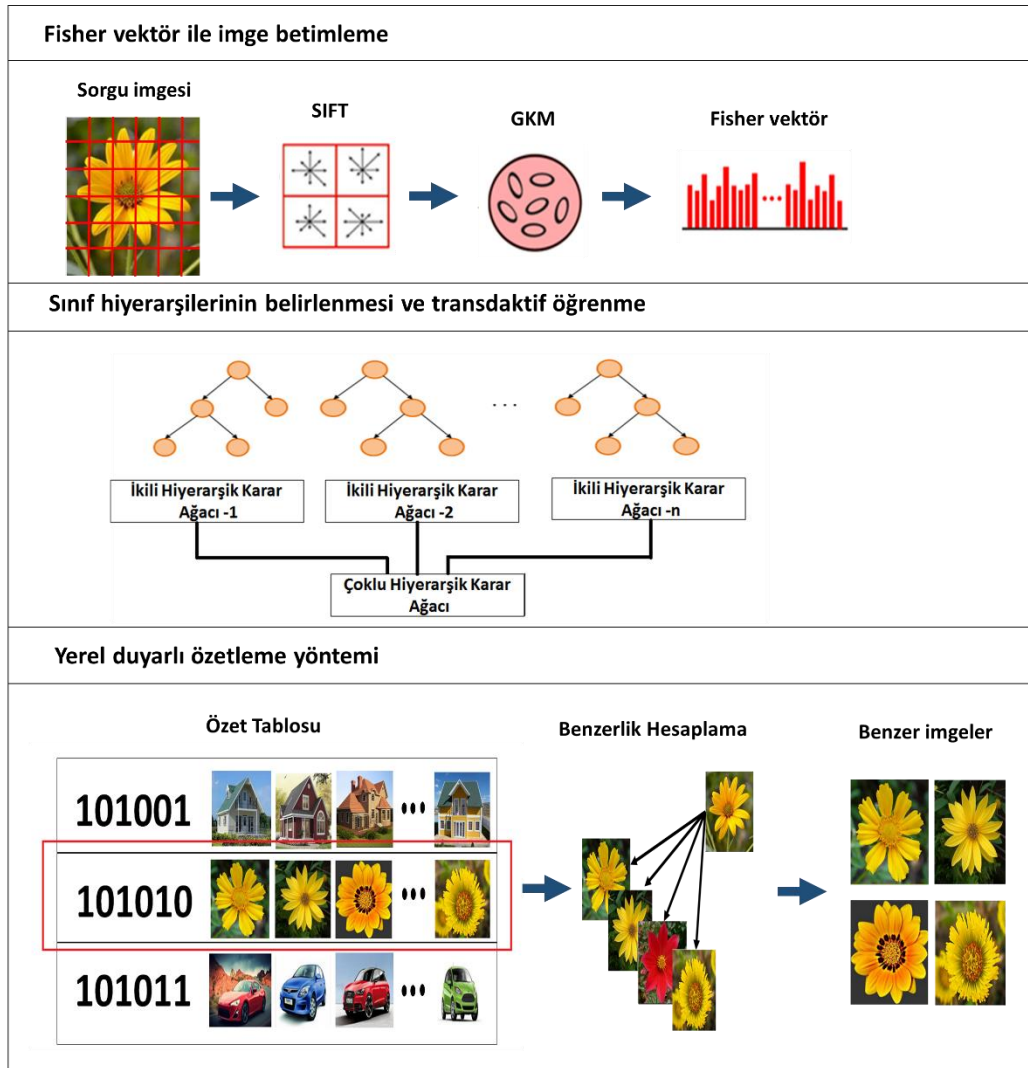
#### 6.4. Yerel Duyarlı Özetleme Yöntemine Dair Deneyler

İmge erişiminde, Fisher vektör gibi yöntemler kullanılarak imge betimlemesi hem işlem hem de saklama yükünü büyük ölçüde azaltmaktadır. Ancak görsel kelime sayısının ya da veri tabanındaki imge sayısının fazla olduğu durumlarda Fisher vektör ile imge betimleme tek başına bu sorunları çözmede yeterli olamamaktadır. Bu nedenle imge histogramlarını daha etkili bir şekilde hafızada saklayabilmek için özetleme işlemi yapılmaktadır. Bu şekilde her bir imge  $b$ -bitten oluşan özet vektörleri ile ifade edilmektedir.

Bir önceki aşamalarda Fisher vektör ile betimlenen imgelerin transdaktif öğrenme ve ikili sınıf hiyerarşik karar ağaçları yöntemleri birlikte kullanılarak ayırt-edici izdüşüm vektörleri bulunmuştur. Ardından bu izdüşüm vektörleri verilerin boyutlarını indirgemedi kullanılmıştır. Boyutu indirgenen verilerin Yerel Duyarlı Özetleme yöntemi kullanılarak özet kodları elde edilmiştir. Elde edilen özet kodlar kullanılarak  $k$ -en yakın komşu sınıflandırıcısının performansı değerlendirilmiştir (Şekil 6.3).

Öncelikle her bir imgeden SIFT uygulanarak çıkarılan öznelik vektörleri kullanılarak,  $\{w_k, \mu_k, \sigma_k\}_k$  sırasıyla tüm imgeleri tanımlamak için kullanılan karışım ağırlıkları, ortalamaları ve diyagonal kovaryansları hesaplanmıştır. Böylece kullanılan veri tabanı için gauss sayısı  $K = 128$  veya  $K = 256$  seçilerek GKM modeli elde edilmiştir. Elde edilen GKM modeli kullanılarak her bir imge Fisher vektör olarak betimlenmiştir. Deneylerde öznelik vektörlerinin boyutu  $d = 64$  veya  $d = 80$  ve  $K = 128$  veya  $K = 256$  seçilerek FV boyutu ( $2Kd$ ) hesaplanmıştır.

Bu süreç her bir veri tabanı için tekrarlanarak her biri için GKM model ayrı ayrı oluşturulmuştur. Ardından tüm veri tabanındaki imgeler Fisher vektörüyle betimlenmiştir. İmgeler betimlendikten sonra birden fazla ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçları oluşturulurken rasgele sınıflar ve örnekler seçilmiştir. Elde edilen karar ağaçlarının düğüm noktalarına transdaktif destek vektör makineleri uygulanarak farklı destek vektör makine sınıflandırıcısı modelleri elde edilmiştir. Destek vektör makine sınıflandırıcısının döndürdüğü hiperdüzlemlerin normal vektörleri veri boyutunu indirgemek için kullanılmıştır.



Şekil 6.3 Yerel duyarlı özetleme yöntemine dair deneyde yapılan aşamalar

Bir sonraki aşamada boyutu indirgenmiş veri kullanılarak lsh komutuyla her bir verinin özet vektörü elde edilerek özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı,  $k$ ), özet tablosu sayısı ( $L$ ) ve izdüşüm genişliği ( $w$ ) veri tabanına en uygun şekilde seçilmiştir. Elde edilen özet tablosu kullanılarak lshlookup komutuyla boyutu indirgenmiş test verisinin, boyutu indirgenmiş eğitim verisi içinden en yakın  $n$  komşusu belirlenmiştir. Bu işlem her bir farklı karar ağacı için tekrarlanmış ve toplamda boyutu indirgenmiş eğitim verisinden en yakın komşular belirlenmiştir. Bu komşulardan tekrarlı olanlar var ise onlar ayıklanmıştır. Boyutu indirgenmiş test verisi ile bulunan en yakın komşular arasındaki uzaklıklar Öklid uzaklık ölçütü kullanılarak her bir karar ağacı için ayrı ayrı hesaplanmıştır. Hesaplanan eğitim ve

test verileri arasındaki uzaklıkların sırasıyla ortalaması, minimumu, çarpımı ve medyanı hesaplanmıştır. Hesaplanan her bir uzaklık ve etiket bilgileri kullanılarak mAP başarımları hesaplanmıştır. Ayrıca bu deneylere ek olarak bulunan en yakın 10 ve 100 komşu için de ayrı ayrı mAP başarımları hesaplanmıştır.

#### 6.4.1. INRIA Holidays veri tabanı üzerindeki deneyler

INRIA Holidays veri tabanı farklı yerler ve nesnelere içeren kişisel tatil fotoğraflarından oluşan bir veri tabanıdır. Bu veri tabanında imgelerin sağlamlığını test etmek amacıyla çeşitli dönüşümler: dönmeler, bakış açısı ve aydınlatma değişiklikleri, bulanıklık, vb uygulanmıştır. Veri tabanı çok çeşitli sahne türleri (doğal, yapay, su, ateş efektleri, vb) içerir ve imgeler yüksek çözünürlüklüdür. Veri tabanı 500 tanesi sorgu imgesi olmak üzere toplamda 1491 imgeden oluşmaktadır. Şekil 6.4'te veri tabanına ait çeşitli örnek imgeler gösterilmektedir. Her grubun ilk imgesi sorgu imgesidir ve bu sorgu imgesine karşılık dönen doğru imgeler bu gruptaki geri kalan imgelerdir.



Şekil 6.4 INRIA Holidays veri tabanına ait çeşitli örnek imgeler (Jegou ve Douze, 2008)

Bu deneyde 500 sınıftan rasgele 80 sınıf seçilmek suretiyle birbirinden farklı 10 tane ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırmak için transdaktif DVM kullanılmıştır. Her bir karar ağacından 79 adet izdüşüm vektörü elde edilmiştir ve böylece veri boyutu ( $K = 256$  ve  $d = 80$  için  $2Kd$ ) 40960'dan  $10 \times 79$ 'a indirgenmiştir. Ardından boyutu indirgenmiş veri kullanılarak özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı)  $k = 50$ , özet tablosu sayısı  $L = 5$  ve izdüşüm genişliği  $w = 0$  olarak seçilmiştir. Elde edilen özet

tablosu kullanılarak lshlookup komutuyla boyutu indirgenmiş test verisinin boyutu indirgenmiş eğitim verisi içinden en yakın 11 komşusu belirlenmiştir. Bu işlem 10 farklı karar ağacı için tekrarlanmış ve toplamda boyutu indirgenmiş eğitim verisinden en yakın 110 komşu belirlenmiştir. Bu komşulardan tekrarlı olanlar var ise onlar ayıklanmıştır. Boyutu indirgenmiş test verisi ile bulunan en yakın komşular arasındaki uzaklıklar Öklid uzaklık ölçütü kullanılarak her bir karar ağacı için ayrı ayrı hesaplanmıştır. Hesaplanan eğitim ve test verileri arasındaki uzaklıkların sırasıyla ortalaması, minimumu, çarpımı ve medyanı hesaplanmıştır. Hesaplanan her bir uzaklık ve etiket bilgileri kullanılarak mAP başarımları hesaplanmıştır. Ayrıca bu deneylere ek olarak bulunan en yakın 10 komşu ve 100 komşu için de ayrı ayrı mAP başarımları hesaplanmıştır. INRIA Holidays veri tabanı üzerindeki deneyin mAP başarımları Çizelge 6.5’de verilmiştir. En yüksek sonuç olan %74,46 uzaklıkların çarpımı hesaplanarak ve en yakın 10 komşu kullanılarak elde edilmiştir. Bu da en yakın komşu sınıflandırıcısı ile elde edilen %71,76’lık sonuca oranla daha iyi bir sonuçtur.

Çizelge 6.5 INRIA Holidays veri tabanı üzerindeki deney sonuçları ( $L = 5, k = 50, w = 0$ )

Yöntem	mAP Başarımı (%)			
	Komşu sayıları	Tüm komşular	En yakın 100 komşu	En yakın 10 komşu
Önerilen Yöntem	Ortalama	69,08	71,72	72,82
	Minimum	62,39	64,19	67,93
	Çarpım	69,66	72,50	<b>74,46</b>
	Medyan	66,95	69,49	72,44
En yakın komşu sınıflandırıcısı	71,76			

INRIA Holidays ve Flickr1M (bu veri tabanı 1 milyon imgeden oluşmaktadır) veri tabanları birlikte kullanılarak yukarıda anlatılan INRIA Holidays veri tabanı için geçerli işlemler ufak değişikliklerle tekrarlanarak farklı bir deneysel çalışma yapılmıştır. Öncelikle Flickr1M veri tabanındaki imgeler Fisher vektör yöntemiyle betimlenmiştir. Ardından 500 sınıftan rasgele 80 sınıf seçilmek suretiyle birbirinden farklı 25 tane ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırmak için

transdaktif DVM kullanılmıştır. Burada Flickr1M veri tabanından rasgele seçilen veriler transdaktif DVM’de etiketsiz veri olarak kullanılmıştır. Her bir karar ağacından 79 adet izdüşüm vektörü elde edilmiştir ve böylece veri boyutu ( $K = 256$  ve  $d = 80$  için  $2Kd$ ) 40960’dan  $25 \times 79$ ’a indirgenmiştir. Ardından boyutu indirgenmiş veri kullanılarak özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı)  $k = 15$ , özet tablosu sayısı  $L = 10$  ve izdüşüm genişliği  $w = 0$  olarak seçilmiştir. Elde edilen özet tablosu kullanılarak boyutu indirgenmiş test verisinin boyutu indirgenmiş eğitim verisi içinden en yakın 11 komşusu belirlenmiştir. Bu işlem 25 farklı karar ağacı için tekrarlanmış ve toplamda boyutu indirgenmiş eğitim verisinden en yakın 275 komşu belirlenmiştir. Daha sonra INRIA Holidays veri tabanı için geçerli işlemler tekrarlanmıştır. Ayrıca bu deneylere ek olarak bulunan en yakın 100 ve 10 komşu için de ayrı ayrı mAP başarımları hesaplanmıştır. Önerilen yöntem ile en yakın komşu sınıflandırıcısıyla elde edilen sonuçlara yakın sonuçlar elde edilmiştir. INRIA Holidays +Flickr1M veri tabanı üzerindeki deneyin mAP başarımları Çizelge 6.6’da verilmiştir.

Çizelge 6.6 INRIA Holidays +Flickr1M veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 15, w = 0$ )

Yöntem	mAP Başarımı (%)			
	Komşu sayıları	Tüm komşular	En yakın 100 komşu	En yakın 10 komşu
Önerilen Yöntem	Ortalama	35,22	40,07	42,61
	Minimum	40,04	45,86	48,56
	Çarpım	38,43	43,45	46,23
	Medyan	43,14	48,71	<b>51,43</b>
En yakın komşu sınıflandırıcısı	57,61			

#### 6.4.2. Kentucky veri tabanı üzerindeki deneyler

Kentucky veri tabanı (Nister ve Stewenius, 2006) 2550 farklı nesne veya sahne içerir. Her biri dört farklı bakış açısından alınan dört imge ile temsil edilmektedir. Bunun sonucu olarak veri tabanı 10200 imge içermektedir. Tüm imgeler sorgu imgesi olarak kullanılabilir. Şekil 6.5’de veri tabanına ait çeşitli örnek imgeler gösterilmektedir.



Şekil 6.5 Kentucky veri tabanına ait çeşitli örnek imgeler (Nister ve Stewenius, 2006).

Bu deneyde 2550 sınıftan rasgele 200 sınıf seçilmek suretiyle birbirinden farklı 15 tane ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırmak için transdaktif DVM kullanılmıştır. Her bir karar ağacından 199 adet izdüşüm vektörü elde edilmiştir ve böylece veri boyutu ( $K = 256$  ve  $d = 80$  için  $2Kd$ ) 40960'dan  $15 \times 199$ 'a indirgenmiştir. Ardından boyutu indirgenmiş veri kullanılarak özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı)  $k = 10$ , özet tablosu sayısı  $L = 10$  ve izdüşüm genişliği  $w = 4$  olarak seçilmiştir. Elde edilen özet tablosu kullanılarak boyutu indirgenmiş test verisinin boyutu indirgenmiş eğitim verisi içinden en yakın 11 komşusu belirlenmiştir. Bu işlem 15 farklı karar ağacı için tekrarlanmış ve toplamda boyutu indirgenmiş eğitim verisinden en yakın 165 komşu belirlenmiştir. Bu komşulardan tekrarlı olanlar var ise onlar ayıklanmıştır. Boyutu indirgenmiş test verisi ile bulunan en yakın komşular arasındaki uzaklıklar Öklid uzaklık ölçütü kullanılarak her bir karar ağacı için ayrı ayrı hesaplanmıştır. Hesaplanan eğitim ve test verileri arasındaki uzaklıkların sırasıyla ortalaması, minimumu, çarpımı ve medyanı hesaplanmıştır. Hesaplanan her bir uzaklık ve etiket bilgileri kullanılarak mAP başarımları hesaplanmıştır. Kentucky veri tabanı üzerindeki deneyin mAP başarımları Çizelge 6.7'de verilmiştir. En yüksek sonuç olan %63,25 uzaklıkların ortalaması hesaplanarak elde edilmiştir. Bu da en yakın komşu sınıflandırıcısı ile elde edilen %52,10'luk sonuca oranla çok daha iyi bir sonuçtur.

Çizelge 6.7 Kentucky veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 4$ )

Yöntem	mAP Başarımı (%)	
Önerilen Yöntem	Ortalama	63,25
	Minimum	55,73
	Çarpım	61,57
	Medyan	56,28
En yakın komşu sınıflandırıcısı	52,10	

Kentucky ve Flickr1M veri tabanları birlikte kullanılarak yukarıda anlatılan Kentucky veri tabanı için geçerli işlemler ufak değişikliklerle tekrarlanarak farklı bir deneysel çalışma yapılmıştır. 2550 sınıftan rasgele 200 sınıf seçilmek suretiyle birbirinden farklı 15 tane ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırmak için transdaktif DVM kullanılmıştır. Burada Flickr1M veri tabanından rasgele seçilen veriler transdaktif DVM’de etiketsiz veri olarak kullanılmıştır. Her bir karar ağacından 199 adet izdüşüm vektörü elde edilmiştir ve böylece veri boyutu ( $K = 256$  ve  $d = 80$  için  $2Kd$ ) 40960’dan  $15 \times 199$ ’a indirgenmiştir. Ardından boyutu indirgenmiş veri kullanılarak özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı)  $k = 10$ , özet tablosu sayısı  $L = 10$  ve izdüşüm genişliği  $w = 4$  olarak seçilmiştir. Elde edilen özet tablosu kullanılarak boyutu indirgenmiş test verisinin boyutu indirgenmiş eğitim verisi içinden en yakın 11 komşusu belirlenmiştir. Bu işlem 15 farklı karar ağacı için tekrarlanmış ve toplamda boyutu indirgenmiş eğitim verisinden en yakın 165 komşu belirlenmiştir. Daha sonra Kentucky veri tabanı için geçerli işlemler tekrarlanmıştır. Kentucky+Flickr1M veri tabanı üzerindeki deneyin mAP başarımları Çizelge 6.8’de verilmiştir. En yüksek sonuç olan %62,09 uzaklıkların ortalaması hesaplanarak elde edilmiştir. Bu da en yakın komşu sınıflandırıcısı ile elde edilen %52,10’luk sonuca oranla çok daha iyi bir sonuçtur.

Çizelge 6.8 Kentucky+Flickr1M veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 4$ )

Yöntem	mAP Başarımı (%)	
Önerilen Yöntem	Ortalama	62,09
	Minimum	55,17
	Çarpım	60,29
	Medyan	56,10
En yakın komşu sınıflandırıcısı	52,10	

### 6.4.3. UC Merced Land Use veri tabanı üzerindeki deneyler

UC Merced Land Use veri tabanı 21 sınıftan (agricultural, airplane, beach, forest, river gibi) oluşmakta ve her bir sınıfa ait  $256 \times 256$  boyutlu 100 imge bulunmaktadır (Şekil 6.6). İmgeler, ülke çapında çeşitli kentsel alanlar için USGS- Ulusal Kentsel Alan Harita Görüntüleri koleksiyonundan büyük imgelerden elle çıkarılmıştır (Yang ve Newsam, 2010).



Şekil 6.6 UC Merced Land Use veri tabanına ait her bir sınıf için örnek imgeler

Yapılan deneylerde 21 sınıftan oluşan veri tabanından her sınıftan 50 imge eğitim, 50 imge sına için seçilmek suretiyle birbirinden farklı 20 tane ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırmak için transdaktif DVM kullanılmıştır. Her bir karar ağacından 20 adet izdüşüm vektörü elde edilmiştir ve böylece veri boyutu ( $K = 128$  ve  $d = 64$  için  $2Kd$ ) 16384'ten  $20 \times 20$ 'a



indirgenmiştir. Ardından boyutu indirgenmiş veri kullanılarak özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı)  $k = 10$ , özet tablosu sayısı  $L = 10$  ve izdüşüm genişliği  $w = 1$  olarak seçilmiştir. Elde edilen özet tablosu kullanılarak boyutu indirgenmiş test verisinin boyutu indirgenmiş eğitim verisi içinden en yakın 250 komşusu belirlenmiştir. Bu işlem 20 farklı karar ağacı için tekrarlanmış ve toplamda boyutu indirgenmiş eğitim verisinden en yakın 5000 komşu belirlenmiştir. Bu komşulardan tekrarlı olanlar var ise onlar ayıklanmıştır. Boyutu indirgenmiş test verisi ile bulunan en yakın komşular arasındaki uzaklıklar Öklid uzaklık ölçütü kullanılarak her bir karar ağacı için ayrı ayrı hesaplanmıştır. Hesaplanan eğitim ve test verileri arasındaki uzaklıkların sırasıyla ortalaması, minimumu, çarpımı ve medyanı hesaplanmıştır. Hesaplanan her bir uzaklık ve etiket bilgileri kullanılarak mAP başarımları hesaplanmıştır. UC Merced Land Use veri tabanı üzerindeki deneyin mAP başarımları Çizelge 6.9’da verilmiştir. En yüksek sonuç olan %78,18 uzaklıkların medyanı hesaplanarak elde edilmiştir. Bu da en yakın komşu sınıflandırıcısı ile elde edilen %30,33’lük sonuca oranla çok daha iyi bir sonuçtur.

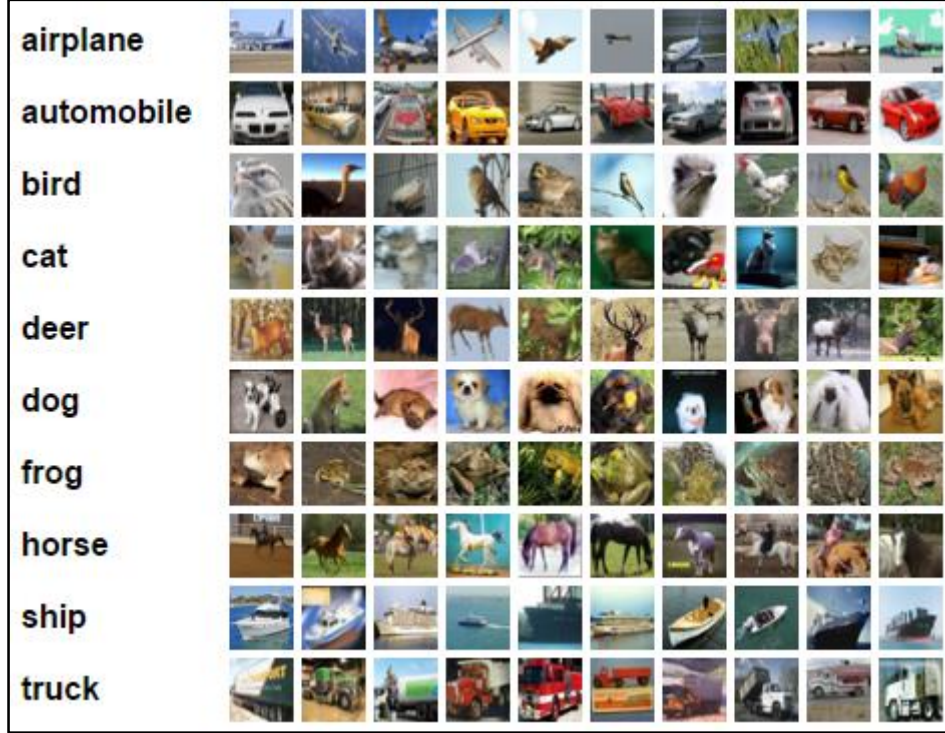
Çizelge 6.9 UC Merced Land Use veri tabanı üzerindeki deney sonuçları ( $L = 10$ ,  $k = 10$ ,  $w = 1$ )( $n = 250$ )

Yöntem	mAP Başarımı (%)	
Önerilen Yöntem	Ortalama	76,40
	Minimum	70,33
	Çarpım	76,65
	Medyan	<b>78,18</b>
En yakın komşu sınıflandırıcısı	30,33	

#### 6.4.4. CIFAR-10 veri tabanı üzerindeki deneyler

CIFAR-10 veri tabanı 10 sınıftan (airplane, automobile, bird, cat, deer, dog, frog, horse, ship ve truck) oluşan toplamda 60000 tane  $32 \times 32$ ’lik renkli imgeden oluşmaktadır. Her sınıfta 6000 imge bulunmaktadır. Bunlardan 50000 tanesi eğitim için, 10000 tanesi sına için ayrılmıştır. Veri tabanı her biri 10000 imgeden oluşacak şekilde, beş eğitim parçası ve bir sına parçası olmak üzere 6 parçaya ayrılmıştır. Sına parçası

her sınıftan rasgele seçilmiş tam olarak 1000 imgeyi içermektedir. Eğitim parçaları kalan imgelerden sırayla rasgele seçilmiş imgeleri içermektedir. Fakat bazı eğitim parçaları bir sınıftan diğerlerine göre daha fazla imge içerebilir. Böylelikle eğitim parçaları her sınıftan tam olarak 5000 tane imgeyi içermektedir (Krizhevsky ve Hinton, 2009). Şekil 6.7da CIFAR-10 veri tabanına ait örnek imgeler gösterilmektedir.



Şekil 6.7 CIFAR-10 veri tabanına ait örnek imgeler

Yapılan deneylerde 10 sınıftan oluşan veri tabanından her bir eğitim parçasından her sınıf için 800 imge eğitim, 800 imge de sınama olarak kullanılmak üzere seçilmiştir. Bunlar kullanılarak birbirinden farklı 10 tane ikili sınıf hiyerarşik karar ağacı oluşturulmuştur. Bu karar ağaçlarının düğümlerindeki verileri ayırmak için transdaktif DVM kullanılmıştır. Her bir karar ağacından 9 adet izdüşüm vektörü elde edilmiştir ve böylece veri boyutu ( $K = 128$  ve  $d = 64$  için  $2Kd$ ) 16384'ten  $10 \times 9$ 'a indirgenmiştir. Ardından boyutu indirgenmiş veri kullanılarak özet tablosu oluşturulmuştur. Özet tablosu oluşturulurken izdüşüm sayısı (bit sayısı)  $k = 10$ , özet tablosu sayısı  $L = 10$  ve izdüşüm genişliği  $w = 0$  olarak seçilmiştir. Elde edilen özet tablosu kullanılarak boyutu indirgenmiş test verisinin boyutu indirgenmiş eğitim verisi içinden en yakın 500 komşusu belirlenmiştir. Bu işlem 10 farklı karar ağacı için tekrarlanmış ve toplamda boyutu

indirgenmiş eğitim verisinden en yakın 5000 komşu belirlenmiştir. Bu komşulardan tekrarlı olanlar var ise onlar ayıklanmıştır. Boyutu indirgenmiş test verisi ile bulunan en yakın komşular arasındaki uzaklıklar Öklid uzaklık ölçütü kullanılarak her bir karar ağacı için ayrı ayrı hesaplanmıştır. Hesaplanan eğitim ve test verileri arasındaki uzaklıkların sırasıyla ortalaması, minimumu, çarpımı ve medyanı hesaplanmıştır. Hesaplanan her bir uzaklığın en yakın 500 tanesi ve etiket bilgileri kullanılarak mAP başarımları hesaplanmıştır. Bu işlemler beş eğitim parçası için de tekrarlanmıştır. Ardından elde edilen başarımların ortalaması alınmıştır. CIFAR-10 veri tabanı üzerindeki deneyin mAP başarımları Çizelge 6.10'da verilmiştir. En yüksek sonuç olan %50,45 uzaklıkların ortalaması hesaplanarak elde edilmiştir. Bu da en yakın komşu sınıflandırıcısı ile elde edilen %26,41'lik sonuca oranla çok daha iyi bir sonuçtur.

Çizelge 6.10 CIFAR-10 veri tabanı üzerindeki deney sonuçları ( $L = 10, k = 10, w = 0$ )( $n = 500$ )

Yöntem	mAP Başarımı (%)	
Önerilen Yöntem	Ortalama	50,45
	Minimum	44,34
	Çarpım	50,10
	Medyan	49,20
En yakın komşu sınıflandırıcısı	26,41	

## 7. SONUÇ VE ÖNERİLER

Tez kapsamında büyük veri tabanları kullanılarak etkin ve başarılı bir imge erişim sistemi geliştirilmesi hedeflenmiştir. Literatürde bu amaç doğrultusunda önerilen birçok yöntem mevcuttur. Bu yöntemler arasında gerçekçi ve büyük veri tabanları kullanılarak geliştirilen yöntemler genellikle etiket bilgisini kullanmamaktadırlar. Ayrıca bu yöntemlerin uzaklık ölçütleri kullanarak giriş örnek uzayındaki uzaklıklara bakıp bu uzaklıkları yansıtan özet kodları üretme üzerine kurulu bir yapısı vardır. Ancak düşük seviye öznitelikler kullanılarak oluşturulan imge öznitelik vektörlerini kullanan uzaklık ölçütü her ne kadar başarılı olursa olsun genellikle imgelerdeki anlam bilimsel ilişkileri yansıtamamaktadır (anlam bilimsel boşluk) ve bundan dolayı bu tür yöntemlerin başarı oranı oldukça düşüktür. Etiket bilgisini kullanan yöntemler ise ya eğitim kümesindeki sınırlı sayıda örneği kullanarak birçok verinin bilgisini kullanmazlar ya da gerçekçi veri tabanları yerine giriş uzay boyutu küçük ve gerçekçi olmayan veri tabanları kullanırlar. Bunun sebebi ise yöntemlerin sadece örnek uzay boyutunun (imgelere ait öznitelik vektörlerinin boyutu) küçük olduğu durumlarda çalışabilmeleridir.

Bu tez çalışmasında hedeflenen, çeşitli yaklaşımlar kullanarak etiket bilgilerini etkin bir şekilde kullanan imge erişim sistemi geliştirmek ve bu sistemi gerçekçi imge veri tabanları üzerinde sınamaktır. Geliştirilen imge erişim sistemindeki en önemli aşamalar şunlardır: (i) imge betimleme, (ii) etiketlerin temizlenmesi, (iii) sınıf hiyerarşilerinin oluşturulması ve (iv) transdaktif destek vektör makineleri ile NCuts kullanılarak özet kodlarının elde edilmesidir. Öncelikle kullanılan imge veri tabanındaki yarı-otomatik algoritmalar tarafından etiketlenen imgelerin hatalı olabilecek etiket bilgilerinin düzeltilmesi üzerinde çalışılmıştır. Bu amaçla yarı-otomatik bir etiketleme aracı geliştirilmiştir. İmgeler Fisher vektör yöntemi ile betimlenmiş, ardından da imgeler ve imgelerin etiket bilgisi kullanılarak sınıf hiyerarşisi oluşturulmuştur. Daha sonra transdaktif öğrenme ve büyük veri tabanlarına uygun yarı-eğitmenli toplama algoritmaları kullanılarak etiket bilgisi olmayan veriler etkin bir şekilde etiketlenmiştir. Ardından anlam bilimsel etiket bilgilerini yansıtan özet kodları oluşturulmuştur. Birçok büyük veri tabanı kullanılarak yapılan deneysel çalışmalarda geliştirilen imge erişim

sistemi sınanmıştır ve sonuç olarak etiket bilgilerini en etkin şekilde kullanan bir imge erişim sistemi geliştirilmiştir.

Deneysel çalışmalar; Pascal VOC 2007, Caltech 256, AR, UC Merced Land Use, CIFAR-10, INRIA Holidays ve Kentucky veri tabanı gibi farklı, etiketli veri tabanları üzerinde gerçekleştirilmiştir. Yapılan deneyler sırasıyla Fisher vektör ile imgelerin betimlenmesi, sınıf hiyerarşilerinin belirlenmesi, transdaktif öğrenme ve yerel duyarlı özetleme yöntemi ile ilgili deneylerdir. Fisher vektör ile imgelerin betimlenmesine dair deneyde; PASCAL VOC 2007 veri tabanı Fisher vektör ile betimlenmiş ve sadece doğrusal destek vektör makineleri kullanılarak PASCAL VOC 2007 yarışmasında elde edilen sonuçlara benzer sonuçlar elde edilmiştir. Böylelikle Fisher vektör kodlarının düzgün bir şekilde çalıştığı ve imge erişim amaçlı kullanılabilceği gözlenmiştir. Sınıf hiyerarşilerinin belirlenmesine dair deneyde; ikili sınıf hiyerarşik karar ağaçları yöntemin başarımı Caltech 256 veri tabanı üzerinde test edilmiştir. Önerilen yöntem en-yakın komşu sınıflandırıcısına oranla çok daha iyi sonuçlar verirken bire-karşı-tüm yaklaşımını kullanan Destek Vektör Makinelerine oranla çok daha kötü sonuçlar vermiştir. Bunları göz önüne alarak başarımı arttırmak için tek bir hiyerarşi oluşturmak yerine birden fazla hiyerarşi oluşturulmuş ve bu yaklaşımın sınıflandırma başarısını önemli ölçüde arttırdığı gözlenmiştir. Transdaktif öğrenmeye dair deneyde; transdaktif destek vektör makinesi yönteminin sınıflandırma başarısını geliştirmek üzere Caltech-256 ve AR yüz veri tabanları üzerinde çalışmalar yapılmıştır. Transdaktif destek vektör makineleri ile ikili karar ağaçları yöntemi birlikte kullanılarak boyut indirgeme suretiyle en yakın komşu sınıflandırma başarısı büyük ölçüde arttırılmıştır. Buna ek olarak boyut azaltıldığından sınıflandırma için geçecek zamanda da önemli kazanımlar elde edildiği gözlemlenmiştir. İmge erişim deneylerinde ise benzer şekilde önerilen yöntem kullanılarak yapılan boyut indirgeme sonrası orjinal giriş uzayını kullanan Öklid uzaklığının performansı önemli ölçülerde arttırılmıştır. Bu sebeple yöntem direkt olarak özet kodlarını oluşturmada kullanılabilceği gibi boyut indirgindikten sonra yerel duyarlı özetleme yöntemi ile birlikte de kullanılabilir. Bu çalışmada önerilen yöntemin başarımı sınıflar arasındaki marjinle yakından alakalıdır. Sınıfların iyi ayrıştırılabildiği ve sınıflar arası marjinin yüksek olduğu durumlarda yöntemin başarımı da yüksek olacaktır. Bu sebeple yöntemi çok fazla veriyle kullanırken sınıfların iyi ayrıştırılabilmesi için Fisher vektör gibi yüksek boyutlu imge özniteliklerine ihtiyaç vardır.

GIST gibi küçük boyutlu öznitelik vektörleri için sınıflar iyi ayrıştırılmadığından önerilen yöntem bu tür durumlarda başarısız olacaktır.

## KAYNAKLAR DİZİNİ

- Ahonen, T., Hadid, A., ve Pietikainen, M., 2006, Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12), 2037-2041.
- Andoni, A., ve Indyk, P., 2005, E 2 LSH 0.1 User Manual.
- Andoni, A., ve Indyk, P., 2006, October, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on* (s. 459-468). IEEE.
- Andoni, A. ve Indyk, P., 2004, E2lsh: Exact Euclidean locality sensitive hashing. <http://www.mit.edu/~andoni/LSH/>.
- Bannour, H., 2013, *Building and Using Knowledge Models for Semantic Image Annotation* (Doctoral dissertation, Châtenay-Malabry, Ecole centrale de Paris).
- Bay, H., Tuytelaars, T., ve Van Gool, L., 2006, Surf: Speeded up robust features. In *Computer vision–ECCV 2006* (s. 404-417). Springer Berlin Heidelberg.
- Bennett, K., ve Demiriz, A., 1999, Semi-supervised support vector machines. *Advances in Neural Information processing systems*, 368-374.
- Brown, M., ve Lowe, D. G., 2002, September, Invariant Features from Interest Point Groups. In *BMVC* (No. s 1).
- Çevikalp, H., Kurt, Z., ve Onarcan, A. O. Fourier Dönüşüm Tabanlı Betimleyici Kullanarak Görsel Nesne Sınıflandırma Return of the King: The Fourier Transform Based Descriptor for Visual Object Classification.
- Çevikalp, H., Verbeek, J., Jurie, F., ve Klaser, A., 2008, Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *VISAPP'08-3rd International Conference on Computer Vision Theory and Applications* (Cilt. 1, s. 489-496). INSTICC.
- Çevikalp, H., 2010, New clustering algorithms for the support vector machine based hierarchical classification. *Pattern Recognition Letters*, 31(11), 1285-1291.
- Çevikalp, H., Triggs, B., Yavuz, H. S., Küçük, Y., Küçük, M., ve Barkana, A., 2010, Large margin classifiers based on affine hulls. *Neurocomputing*, 73(16), 3160-3168.
- Chatfield, K., Lempitsky, V. S., Vedaldi, A., ve Zisserman, A., 2011, September, The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC* (Cilt. 2, No. 4, s. 8).

### KAYNAKLAR DİZİNİ (devam)

- Collobert, R., Sinz, F., Weston, J., ve Bottou, L., 2006, Large scale transductive SVMs. *The Journal of Machine Learning Research*, 7, 1687-1712.
- Cortes, C., ve Vapnik, V., 1995, Support-vector networks. *Machine learning*, 20(3), 273-297.
- Csurka, G., Dance, C., Fan, L., Willamowski, J., ve Bray, C., 2004, May, Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV (Cilt. 1, No. 1-22, s. 1-2)*..
- Datar, M., Immorlica, N., Indyk, P., ve Mirrokni, V. S., 2004, June, Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry (s. 253-262)*. ACM.
- Döngel, T., ve Çevikalp, H., 2014, April, Semi automatic annotation tool. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd (s. 1067-1070)*. IEEE.
- Döngel, T., ve Çevikalp, H., 2015, May, Dimensionality reduction by using transductive learning and binary hierarchical trees. In *Signal Processing and Communications Applications Conference (SIU), 2015 23th (s. 767-770)*. IEEE.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., ve Zisserman, A., 2010, The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303-338.
- Fergus, R., Fei-Fei, L., Perona, P., ve Zisserman, A., 2005, October, Learning object categories from Google's image search. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on (Cilt 2, s. 1816-1823)*. IEEE.
- Fergus, R., Bernal, H., Weiss, Y., ve Torralba, A., 2010, Semantic label sharing for learning with many categories. In *Computer Vision–ECCV 2010 (s. 762-775)*. Springer Berlin Heidelberg.
- Gionis, A., Indyk, P., ve Motwani, R., 1999, September, Similarity search in high dimensions via hashing. In *VLDB (Cilt. 99, No. 6, s. 518-529)*.
- Grauman, K., ve Fergus, R., 2013, Learning binary hash codes for large-scale image search. In *Machine learning for computer vision (s. 49-87)*. Springer Berlin Heidelberg.
- Grauman, K., ve Leibe, B., 2009, Visual Recognition. Excerpt chapter from *Synthesis lecture draft: Visual Recognition*
- Griffin, G., Holub, A., ve Perona, P., 2007, Caltech-256 object category dataset.



### KAYNAKLAR DİZİNİ (devam)

- Harzallah, H., Jurie, F., ve Schmid, C., 2009, September, Combining efficient object localization and image classification. In *Computer Vision, 2009 IEEE 12th International Conference on* (s. 237-244). IEEE.
- Indyk, P., ve Motwani, R., 1998, May, Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (s. 604-613). ACM.
- Inria Lear, 2012, Fisher vector implementation, INRIA's Fisher vector implementation, [http://lear.inrialpes.fr/src/inria\\_Fisher/](http://lear.inrialpes.fr/src/inria_Fisher/).
- Jaakkola, T., ve Haussler, D., 1999, Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, 487-493.
- Jegou, H., ve Douze, M., 2008 INRIA Holidays dataset. <http://lear.inrialpes.fr/people/jegou/data.php>.
- Jegou, H., ve Douze, M. INRIA Flickr60k and Flickr1M datasets <http://bigimbas.inrialpes.fr/herve/Flickr60K/>, <http://bigimbas.inrialpes.fr/herve/flickr1M/>
- Joachims, T., 1999, June, Transductive inference for text classification using support vector machines. In *ICML (Cilt 99, s. 200-209)*.
- Jurie, F., ve Triggs, B., 2005, October, Creating efficient codebooks for visual recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Cilt. 1, s. 604-610). IEEE.
- Koenderink, J. J., 1984, The structure of images. *Biological cybernetics*,50(5), 363-370.
- Koniusz, P., ve Mikolajczyk, K., 2010, August, On a quest for image descriptors based on unsupervised segmentation maps. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (s 762-765). IEEE.
- Kristen Grauman ve Bastian Leibe. *Visual Recognition. Excerpt chapter from Synthesis lecture draft: Visual Recognition.*
- Krizhevsky, A., ve Hinton, G., 2009, Learning multiple layers of features from tiny images.
- Kulis, B., ve Grauman, K., 2009, September, Kernelized locality-sensitive hashing for scalable image search. In *Computer Vision, 2009 IEEE 12th International Conference on* (s. 2130-2137). IEEE.

### KAYNAKLAR DİZİNİ (devam)

- Lazebnik, S., Schmid, C., ve Ponce, J., 2006, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (Cilt. 2, s. 2169-2178). IEEE
- Lindeberg, T., 1994, Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2), 225-270.
- Liu, W., Wang, J., Kumar, S., ve Chang, S. F., 2011, Hashing with graphs. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (s. 1-8).
- Lowe, D. G., 2004, Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- Lowe, D. G., 1999, Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Cilt. 2, s. 1150-1157). IEEE.
- Mikolajczyk, K., ve Schmid, C., 2004, Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1), 63-86.
- Mikolajczyk, K., ve Schmid, C. 2002. An affine invariant interest point detector. In *Computer Vision—ECCV 2002* (s. 128-142). Springer Berlin Heidelberg.
- Mikolajczyk, K., ve Schmid, C., 2005, A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10), 1615-1630.
- Nister, D., ve Stewenius, H., 2006, Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (Cilt. 2, s. 2161-2168). IEEE.
- Nowak, E., Jurie, F., ve Triggs, B., 2006, Sampling strategies for bag-of-features image classification. In *Computer Vision—ECCV 2006* (s. 490-503). Springer Berlin Heidelberg.
- Ojala, T., Pietikäinen, M., ve Harwood, D., 1996, A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1), 51-59.
- Perronnin, F., Sánchez, J., ve Mensink, T., 2010, Improving the Fisher kernel for large-scale image classification. In *Computer Vision—ECCV 2010* (s. 143-156). Springer Berlin Heidelberg.

### KAYNAKLAR DİZİNİ (devam)

- Perronnin, F., ve Dance, C., 2007, June, Fisher kernels on visual vocabularies for image categorization. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (s. 1-8). IEEE.
- Platt, J., 1999, Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods—support vector learning, 3.
- Raginsky, M., ve Lazebnik, S., 2009, Locality-sensitive binary codes from shift-invariant kernels. In Advances in neural information processing systems(s. 1509-1517).
- Rahimi, A., ve Recht, B., 2004, Clustering with normalized cuts is clustering with a hyperplane. Statistical Learning in Computer Vision, 56.
- Salakhutdinov, R., ve Hinton, G., 2009, Semantic hashing. International Journal of Approximate Reasoning, 50(7), 969-978.
- Schmid, C., ve Mohr, R., 1997, Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5), 530-534.
- Shakhnarovich, G., 2009, Matlab Toolbox – E2LSH  
<http://ttic.uchicago.edu/~gregory/download.html>
- Simonyan, K., Parkhi, O., ve Vedaldi, A., 2013, Fisher vector faces in the wild, British Machine Vision Conference.
- Sivic, J., ve Zisserman, A., 2003, October, Video Google: A text retrieval approach to object matching in videos. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on (s. 1470-1477). IEEE.
- Tan, X., ve Triggs, B., 2010, Enhanced local texture feature sets for face recognition under difficult lighting conditions. Image Processing, IEEE Transactions on, 19(6), 1635-1650.
- Torralba, A., Fergus, R., ve Weiss, Y., 2008, June, Small codes and large image databases for recognition. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (s. 1-8). IEEE.
- Verbeek, J., 2012 Slides: Fisher vector image representation.
- Villegas, M., Paredes, R., ve Thomee, B., 2013, September, Overview of the imageclef 2013 scalable concept image annotation subtask. In CLEF 2013 Evaluation Labs and Workshop, Online Working Notes, Valencia, Spain.
- Wang, J., Kumar, S., ve Chang, S. F., 2010, June, Semi-supervised hashing for scalable image retrieval. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (s. 3424-3431). IEEE.

**KAYNAKLAR DİZİNİ (devam)**

- Weiss, Y., Torralba, A., ve Fergus, R., 2009, Spectral hashing. In Advances in neural information processing systems (s. 1753-1760).
- Yang, Y., ve Newsam, S., 2010, November, Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (s. 270-279). ACM.
- Zolotarev, V. M., 1986, One-dimensional stable distributions (Cilt 65). American Mathematical Soc..