

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**VERİMLİ NEREDEYSE ÖZDEŞ ERİŞİMİ İÇİN AYIRICI  
ÖZNİTELİK SEÇİMİ**

**YÜKSEK LİSANS TEZİ**  
**Burak YILDIZ**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Doç. Dr. M. Fatih DEMİRCİ**

**Nisan 2016**



Fen Bilimleri Enstitüsü Onayı

.....  
**Prof. Dr. Osman EROĞUL**  
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onayla-  
rım.

.....  
**Prof. Dr. Murat ALANYALI**  
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün **121111111** numaralı Yüksek Li-  
sans öğrencisi **Burak YILDIZ**'ın ilgili yönetmeliklerin belirlediği gerekli  
tüm şartları yerine getirdikten sonra hazırladığı "**VERİMLİ NEREDEYSE  
ÖZDEŞ ERİŞİMİ İÇİN AYIRICI ÖZNİTELİK SEÇİMİ**" başlıklı tezi  
**07 Nisan 2016** tarihinde aşağıda imzaları olan jüri tarafından kabul edil-  
miştir.

**Tez Danışmanı:** **Doç. Dr. M. Fatih DEMİRCİ** .....  
TOBB Ekonomi ve Teknoloji Üniversitesi

**Jüri Üyeleri:** **Yrd. Doç. Dr. Ç. Evren GEREDE**  
**(Başkan)** .....  
TOBB Ekonomi ve Teknoloji Üniversitesi

**Doç. Dr. Fatih EMEKCİ** .....  
Turgut Özal Üniversitesi



## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Burak YILDIZ



## ÖZET

Yüksek Lisans Tezi

### VERİMLİ NEREDEYSE ÖZDEŞ ERİŞİMİ İÇİN AYIRICI ÖZİNİTELİK SEÇİMİ

Burak YILDIZ

TOBB Ekonomi ve Teknoloji Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. M. Fatih DEMİRCİ

Tarih: Nisan 2016

Belirgin öznitelik alt kümesini oluşturma, neredeyse özdeş görüntü erişimi için iki açıdan önemlidir. Bunlardan ilki sorgu süresi önemli ölçüde azalır. İkincisi ise neredeyse özdeş görüntü algılamada belirgin özniteliklerin alt kümesini kullanmak sıradan alt kümeleri kullanmaktan daha iyi sonuç verir. Bu tezde belirgin öznitelik seçimi için özgün bir yöntem önerildi. Bu yöntemde belirgin alt küme, öznitelikleri derecelendirmek için yine bu özniteliklerle hesaplanan bir yoğunluk haritasıyla oluşturuldu. Önerilen bu metodun performansını göstermek için bazı testler yapıldı ve ortalamada %93.08'lik geri çağırmaya karşılık %98.54'lük bir kesinlik elde edildi.

**Anahtar Kelimeler:** Öznitelik seçimi, Neredeyse özdeş görüntü tespiti, Neredeyse özdeş görüntü bulma, Yoğunluk haritası oluşturma.





## ABSTRACT

Master of Science

### DISTINCTIVE FEATURE SELECTION FOR EFFICIENT NEAR-DUPLICATE RETRIEVAL

Burak YILDIZ

TOBB University of Economics and Technology  
Institute of Natural and Applied Sciences  
Department of Computer Engineering

Supervisor: Assoc. Prof. M. Fatih DEMİRÇİ

Date: April 2016

The creation of distinctive subset of the features for near-duplicate image detection is significant in two terms. The former is that the query time decreases reasonably. The latter is that using the distinctive subset of the features for near-duplicate detection performs better than the ordinary subsets. In this thesis, we propose a novel interest point selection method. Using such a method, the distinctive subset is created with a density map, which is calculated from the features to rank them. We examined some experiments to show the performance of the proposed method. The recall rate of 93.08% is obtained when the precision is 98.54% in average.

**Keywords:** Feature selection, Near-duplicate image detection, Near-duplicate image retrieval, Density map creation.



## TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. M. Fatih DEMİRCİ'ye; kıymetli tecrübelerinden faydalandığım üniversitem TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendislięi öğretim üyelerine; kurumum Turgut Özal Üniversitesi Bölümü öğretim üyelerine; bu tezi tamamlamam hususunda desteklerini esirgemeyen Doç. Dr. Fatih EMEKÇİ'ye; bilgisayar mühendislięi alanında yüksek lisansa başlamamda beni destekleyen Yrd. Doç. Dr. Ç. Evren GEREDEYE'ye; tezimi ilk okuyan ve hatalarımı düzelten ve destekleriyle her zaman yanımda olan eşim Kübra'ya; beni bu güne kadar hep destekleyen ve kararlarımda sürekli arkamda olan Annem'e, Babam'a ve Kardeşim'e; sağladığı burs desteęi için TOBB ETÜ'ye; ve burada isimlerini tek tek zikredemediğim hocalarıma ve kardeş bildiğim arkadaşlarıma teşekkürü bir borç bilirim.



## İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET .....	iv
ABSTRACT .....	v
TEŞEKKÜR .....	vi
İÇİNDEKİLER .....	vii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ.....	xi
KISALTMALAR .....	xii
SEMBOL LİSTESİ .....	xiii
RESİM LİSTESİ.....	xiv
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Tezin Amacı .....	2
<b>2. LİTERATÜR ARAŞTIRMASI</b> .....	<b>5</b>
2.1 Filigranla İşaretleme Yöntemiyle Neredeyse Özdeş Tespiti.....	5
2.2 İçerik Tabanlı Görüntü Erişimiyle Neredeyse Özdeş Tespiti .....	5
2.3 Yerel Özniteliklerle Neredeyse Özdeş Tespiti .....	6
2.4 İki Öznitelik Kümesi Arasındaki Benzerliği Ölçmek İçin Kullanılan Yöntemler .....	7
2.4.1 Minkowski-form uzaklık ailesi .....	7
2.4.2 X kare uzaklığı .....	8
2.4.3 Hausdorff uzaklığı .....	8
2.4.4 Toprak taşıyıcı uzaklığı .....	8
<b>3. YOĞUNLUK HARİTASI İLE ÖZİNİTELİK SEÇİMİ</b> .....	<b>11</b>
3.1 Öznitelik Çıkarma .....	13
3.2 Yoğunluk Haritası Oluşturma .....	13
3.3 Öznitelik Seçimi .....	16
<b>4. DENEYSEL ÇALIŞMALAR</b> .....	<b>17</b>
4.1 Başarım Metrikleri .....	17
4.2 Değerlendirme Yöntemi .....	18
4.3 Veri Kümesi.....	20
4.4 Deney Düzenegi .....	25
4.5 Deneysel Sonuçlar .....	26
4.6 Seçilen Öznitelik Sayısının Sonuçlara Etkisi .....	33
4.7 Dönüşüm Türlerinin Sonuçlara Etkisi.....	35
4.8 Sonuçları Etkileyen Diğer Etkenler.....	37

5. SONUÇ VE ÖNERİLER .....	41
KAYNAKLAR.....	43
ÖZGEÇMİŞ .....	47



## ŞEKİL LİSTESİ

### Sayfa

Şekil 1.1: Önerilen yöntemin özeti. (1) Görüntü üzerindeki bütün öznelikler çıkarılır. (2) Öznelikler kullanılarak yoğunluk haritası hesaplanır. (3) Yoğunluk haritasına göre öznelik seçimi yapılır. ....	2
Şekil 2.1: Tepeler ve çukurlardan oluşan örnek bir dağılım.....	9
Şekil 3.1: Örnek bir görüntü için öznelik seçimi (a) Örnek görüntü (b) Tüm SIFT öznelikleri (c) Tüm özneliklerle birlikte örnek görüntü (d) Hesaplanan yoğunluk haritası (e) Seçilen öznelikler (f) Seçilen özneliklerle birlikte örnek görüntü ...	12
Şekil 3.2: Doğru üzerindeki 2 öznelik noktası için yoğunluk haritası oluşturma (a) 3.5 noktasında bulunan birincinin Gaussian katkısı (b) 6.5 noktasında bulunan ikincinin Gaussian katkısı, (c) edilen yoğunluk haritası.....	13
Şekil 3.3: Bir görüntü üzerindeki 2 öznelik noktası için yoğunluk haritası oluşturma (a) 1. özneliğin Gauss katkısı (b) 2. özneliğin Gauss katkısı (c) Yoğunluk haritasının 1. ve 2. özneliklerin Gauss katkılarının toplamları ile oluşmuş hali.....	14
Şekil 4.1: Keskinlik ve Geri Çağırma'nın görsel olarak ifadesi. (Kaynak: İngilizce Wikipedia sayfasından alınarak düzenlenmiş ve türkçeleştirilmiştir).....	17
Şekil 4.2: Bir sorgu görüntüsü için neredeyse özdeşlerin veritabanından bulunma yöntemini anlatan şekil. Bu şekilde koyu renkli yazılmış başlıklar yapılan işlemleri ifade etmektedir. Açık renkliler ise resimdeki nesnelerin adları için kullanılmıştır. ....	19
Şekil 4.3: Görüntü sayısına göre sorgu ve ön işlem zamanı değişimleri (a) Önerilen yöntem ve yöntem 2 için sorgu zamanı-görüntü sayısı grafiği (b) Önerilen yöntem ve yöntem 2 için ön işlem zamanı-görüntü sayısı grafiği .....	29
Şekil 4.4: Yapılan her bir test için kesinlik-geri çağırma grafiği (a) Test 1 için kesinlik-geri çağırma grafiği (b) Test 2 için kesinlik-geri çağırma grafiği (c) Test 3 için kesinlik-geri çağırma grafiği (d) Test 4 için kesinlik-geri çağırma grafiği (e) Test 5 için kesinlik-geri çağırma grafiği (f) Test 6 için kesinlik-geri çağırma grafiği	32

Şekil 4.5: Öznitelik sayısına göre F-değeri ve sorgu zamanı değişimleri (a) F-değerine karşılık öznitelik sayısı grafiği (b) Sorgu zamanına karşılık öznitelik sayısı grafiği .....	34
Şekil 4.6: Dönüşüm türlerine göre en iyi F-değeri için ortalama geri çağırma değerleri .....	37
Şekil 4.7: Dönüşüm türlerine göre en iyi F-değeri için ortalama keskinlik değerleri .....	37





## ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 4.1: Deneysel sonuçlar .....	31





## KISALTMALAR

<b>SIFT</b>	: Ölçekten Bağımsız Öznitelik Dönüşümü (Scale Invariant Feature Transform)
<b>PCA-SIFT</b>	: SIFT için Temel Bileşen Analizi (Principle Component Analysis of SIFT)
<b>SURF</b>	: Hızlandırılmış Dayanıklı Öznitelikler (Speeded Up Robust Features)
<b>FAST</b>	: Hızlandırılmış Parça Testinden Öznitelikler (Features From Accelerated Segment Test)
<b>LSH</b>	: Konum Duyarlı Özetleme (Locality Sensitive Hashing)
<b>RANSAC</b>	: Rastgele Örneklem Birliği (Random Sample Consensus)
<b>DoG</b>	: Gaussian Farkları (Difference of Gaussian)
<b>EMD</b>	: Toprak Taşıyıcı Uzaklığı (Earth Mover's Distance)
<b>FastEMD</b>	: Hızlı Toprak Taşıyıcı Uzaklığı (Fast Earth Mover's Distance)
<b>HDFS</b>	: Hadoop Dağıtık Dosya Sistemi (Hadoop Distributed File System)



## SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

<b>Simgeler</b>	<b>Açıklama</b>
$L_p$	Minkowski-form uzaklık ailesi
$L_1$	Manhattan uzaklığı
$L_2$	Öklit uzaklığı
$X^2$	X kare uzaklığı
$d_H$	Hausdorff uzaklığı
$f_{i,j}$	$i$ 'den $j$ 'ye akış
$F = [f_{i,j}]$	F akışı
$C_{min}$	Minimum maliyet
$\hat{p}(x, y)$	$(x, y)$ noktasındaki yoğunluk
$\mu$	Gauss beklenen değeri
$\sigma^2$	Gauss varyansı
$R$	Yüzde olarak geri çağırma
$P$	Yüzde olarak keskinlik
$F$	F-değeri



## RESİM LİSTESİ

	<u>Sayfa</u>
Resim 4.1: Oluşturulan veri kümesinden 24 örnek görüntü.....	21
Resim 4.2: Örnek bir görüntüye Sayfa 22’de verilen listedeki dönüşümler uygulanarak elde edilen neredeyse özdeş görüntüler soldan sağa ve yukarıdan aşağıya listedeki sırada verilmiştir .....	24
Resim 4.3: Örnek bir görüntü için sorgu görüntüsüyle birlikte sorgu sonucundan sıralı olarak ilk 24 görüntü.....	27
Resim 4.4: Öznitelik sayısı az olan ve orijinalinin %70’i kırılmış bir görüntü için sorgu görüntüsüyle birlikte sorgu sonucundan sıralı olarak ilk 24 görüntü. Yanlış pozitif olanlar kırmızı çerçeveye içine alınmıştır. ....	39

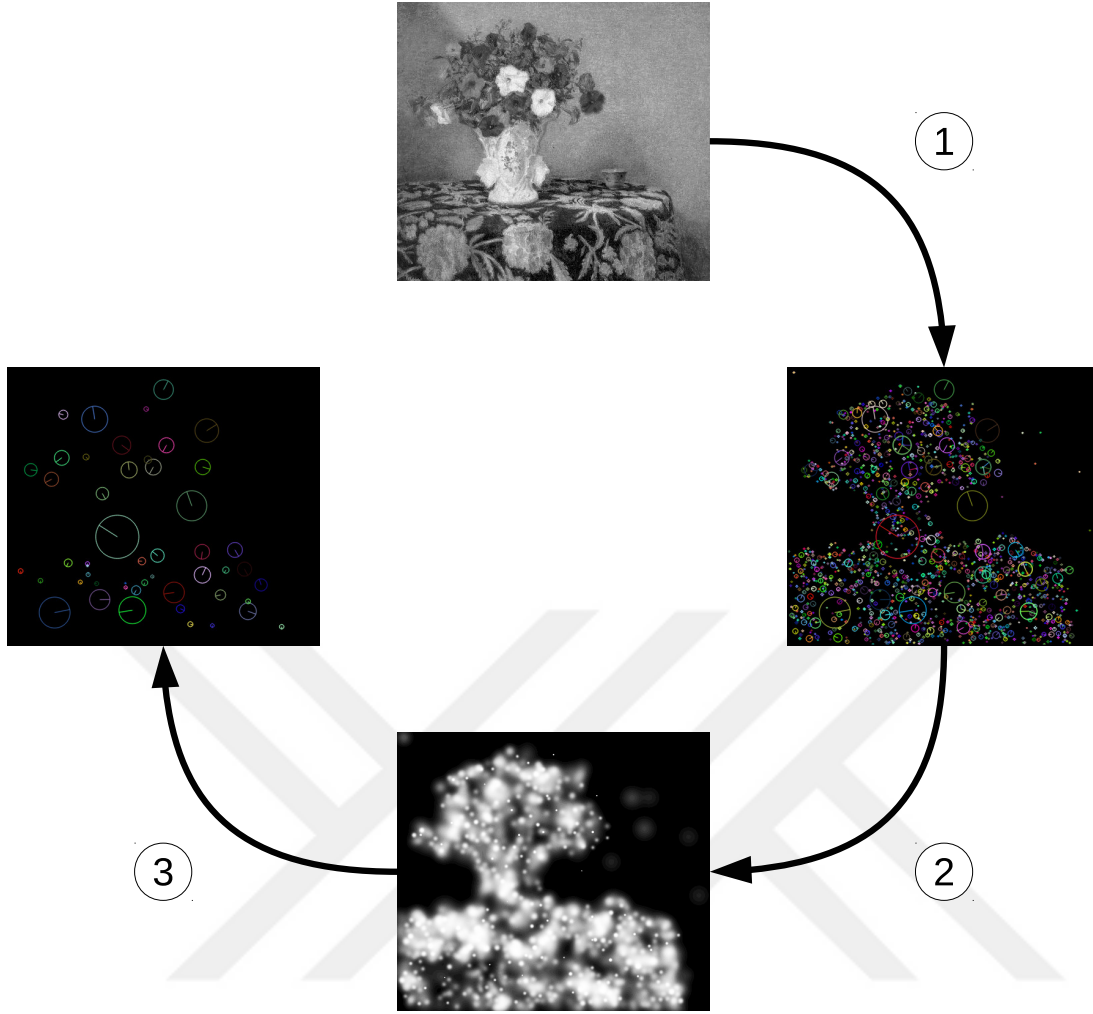




## 1. GİRİŞ

Bilgisayar dünyasındaki gelişmeler çok büyük verilerin hızlı bir şekilde işlenmesine zemin hazırlamıştır. Bu büyük verilerin en başında dijital görsel veriler gelmektedir. Bugün her bir telefonda dahi yüksek çözünürlüklü kameralar yer almakta, her köşe başında bir güvenlik kamerası bulunmaktadır. Trafikte araç içi kameralar sürekli görsel veriler üretmektedir. Bütün bu görsel verilerin bir kısmı özellikle yaygın olarak kullanılan sosyal medyalar dahil her türlü ortamda paylaşılmakta ve yayılmaktadır. Aynı zamanda bugün herhangi bir yolla elde edilen görüntü üzerinde oynamalar yapmak çok kolay hale gelmiştir.

Masaüstü bilgisayarlar için geliştirilen profesyonel programların yanında akıllı telefonlar ve tablet bilgisayarlar için de çok kolaylıkla bir görüntü üzerinde farklı farklı oynamalar yapabilecek ve o üzerinde oynama yapılmış görüntüyü internet aracılığıyla hızlı ve kolay bir şekilde paylaşabilecek uygulamalar geliştirilmiştir. Hayatımıza bir çok alanda hız ve kolaylık getiren bu gelişmeler bazı problemleri de beraberinde getirmiştir. Bunlardan bir tanesi bu tezde ele alınacak olan telif haklarının kolaylıkla çığnenebilir hale gelmesidir. Bu problem, üzerinde telif hakları olan dijital görüntülerin hak sahibinden izinsiz bir şekilde kullanılmasından kaynaklanmaktadır. Yapılan gözlemlerden bu şekilde izinsiz kullanılan görüntülerin büyük bir kısmının, görüntü üzerinde çok az bir değişiklik veya görüntünün bir kısmının kırılarak kullanıldığı görülmüştür. Bu tür görüntüler neredeyse özdeş görüntüler olarak adlandırılmaktadır. Daha bütüncül bir ifadeyle, neredeyse özdeş görüntüler; orijinal görüntülere kesme, döndürme, yeniden boyutlandırma gibi geometrik ve renk değiştirme, kontrast ve parlaklık ayarları ve ışık seviyesi değiştirme gibi fotometrik dönüşümler uygulanarak elde edilir. Bu sebepten dolayı neredeyse özdeş görüntülerin hızlı ve kolay tespiti telif haklarını koruma adına önem taşımaktadır. Bununla beraber, neredeyse özdeş görüntü tespitinin robotik ve video karelerinin geri erişimi gibi birtakım diğer alanlarda da uygulamaları vardır.



Şekil 1.1: Önerilen yöntemin özeti. (1) Görüntü üzerindeki bütün öznitelikler çıkarılır. (2) Öznitelikler kullanılarak yoğunluk haritası hesaplanır. (3) Yoğunluk haritasına göre öznitelik seçimi yapılır.

### 1.1 Tezin Amacı

Neredeyse özdeş görüntüler, ortak bölümler içeren görüntülerden çok daha fazla ortak özniteliğe sahiptir. Bundan dolayı, tüm özniteliklerin küçük bir alt kümesi iki görüntünün neredeyse özdeş tespiti için yeterlidir. Fakat burada bu alt kümenin tüm özniteliklerin içerisinde neredeyse özdeşler içerisinde tekrar etme olasılığı yüksek olanlardan oluşturulması önemlidir. Bu tezde, bu alt kümenin özniteliklerin tekrar olasılığı yüksek olan bölgelerin belirlenerek o bölgelerden seçilen özniteliklerden oluşturulması için yeni bir yöntem önerilmiştir. Bu yöntem ile daha önce çalışması [1] yapılan ışık ve renk değişimlerine karşı sadece ışık değeri temel alınarak seçilen öznitelik alt kümesinden daha dirençli bir öznitelik alt kümesi oluşturulmuştur ve önerilen yöntemin etkinliği yapılan testlerle gösterilmiştir.

Önerilen yöntem Şekil 1.1’de görsel olarak özetlenmiştir. Burada birinci adımda görüntü üzerindeki öznitelikler çıkarılmıştır. İkinci adımda daha sonradan ayrınılı verilecek olan yoğunluk haritası hesaplaması yapılmıştır. Bu yoğunluk haritası, özniteliklerin görüntü üzerindeki konumları ve buna benzer sayısal verileri kullanılarak hesaplanmıştır. Son adımda ise hesaplanan bu yoğunluk haritası kullanılarak özniteliklerin neredeyse özdeş görüntü tespiti için nitelikli bir alt kümesi oluşturulmuştur.





## 2. LİTERATÜR ARAŞTIRMASI

Neredeyse özdeş görüntü tespitinde farklı yöntemler kullanıla gelmiştir. Bu bölümde daha önceden yapılmış bu çalışmalar özetlenecektir. Bu çalışmalara ek olarak literatürde öznitelik kümelerinin benzerliğini ölçmek için kullanılan yöntemlerden bahsedilecektir.

### 2.1 Filigranla İşaretleme Yöntemiyle Neredeyse Özdeş Tespiti

Filigranla işaretleme [2, 3, 4, 5] sıklıkla kullanılan yöntemlerden bir tanesidir. Bu yöntemde görüntüler yayınlanmadan önce içlerine o görüntünün sahibine özel veriler geliştirilmiş algoritmalarla gömülür. Daha sonra işaretlenen bu görüntüler yayımlanır. Bir görüntünün yayınlanan görüntülerden birinin neredeyse özdeşi olup olmadığına, veri gömerken kullanılan algoritmanın tersi bir algoritmayla o görüntüden elde edilen özel veriyle ilk gömülen özel veri karşılaştırılarak karar verilir. Bu yöntem neredeyse özdeş görüntü tespiti için her ne kadar etkili bir çözüm sunsa da, yeterli doğruluk ve keskinliğe sahip olup olmadığı hala tartışmalı bir konudur [6].

### 2.2 İçerik Tabanlı Görüntü Erişimiyle Neredeyse Özdeş Tespiti

İçerik tabanlı görüntü erişimi, neredeyse özdeş görüntü tespitinde kullanılan bir diğer yöntemdir [7, 8, 9, 10, 11, 12, 13]. Bu yöntemde, neredeyse özdeş görüntü bulma işlemi görüntülerden elde edilen global ve yerel istatistiksel veriler kullanılarak yapılır. Fakat global veriler görüntü üzerindeki küçük oynamalardan çok etkilenirler ve bundan dolayı geri çağırma ve keskinlik başarımları düşer. Yerel veriler ise görüntü üzerindeki çok sayıda küçük bölgeden elde edilen istatistiksel verilerin bir vektör haline getirilmesiyle kullanılır. Bundan dolayı görüntüleri eşleştirmek için kullanılan eşik aralığı geniş tutulur. Bu ise normalden fazla doğru görünümü yanlı sonuçların oluşmasına sebep olur.

Son zamanlarda içerik tabanlı görüntü erişimiyle neredeyse özdeş tespiti yapılan çalışmalardan bir tanesi Zheng ve arkadaşlarının yaptığı çalışmadır [12]. Bu

çalışmada Riemannian uzayında görüntü özelliklerinin belirgin bölgelerinin kovaryans matrisini kullanan bir yöntem geliştirilmiştir. Kim ve arkadaşları ise bir diğer içerik tabanlı neredeyse özdeş keşfini beslemeli büyüyen bir yöntemle doğru görünümü yanlışları eleyerek yapan bir uygulama geliştirmişlerdir [13]. Geliştirilen bu uygulama MapReduce [14] arayüzünü kullanarak 250 düğümlü bir makine kümesi üzerinde test edilmiştir. Fakat uygulamanın, global ve yerel görüntü özelliklerinin yazı içerikli resimlerde iyi sonuçlar vermemesinden dolayı, bu tip görüntülerin neredeyse özdeş tespitinde problemlili olduğu görülmüştür.

### 2.3 Yerel Özniteliklerle Neredeyse Özdeş Tespiti

Bu tezde ele alınacak son yöntem neredeyse özdeş görüntü tespitinin yerel öznitelikler kullanılarak yapılmasıdır [15, 1, 16, 17]. Bu yöntemde görüntülerdeki SIFT [18], SURF [19] ve FAST [20] gibi yerel öznitelikler çıkarılarak bu özniteliklerle bir dizin oluşturulur ve daha sonra neredeyse özdeş görüntüler bu dizin üzerinde arama yapılarak bulunur. Ke ve arkadaşları SIFT özniteliğini kullanarak yaptıkları çalışmada [15] 20000 görüntü üzerinde yaptıkları testler sonucunda oldukça iyi sonuçlar elde etmiştir. Bu çalışmada görüntülerden elde edilen SIFT öznitelikleri ve bu özniteliklerin PCA-SIFT [21] tanımlayıcıları LSH [22] yöntemiyle verimli bir dizin yapısına kaydedilmiştir. Neredeyse özdeş görüntü sorgulamaları ise bu verimli dizin yapısı üzerinde yapılmış ve elde edilen sonuçlar geometrik bir doğrulama yöntemi olan RANSAC [23] ile iyileştirilmiştir. Ke ve arkadaşlarının yaptığı bu çalışmadaki önerilen yöntemin kısıtlarından bir tanesi elde edilen SIFT özniteliklerinin sayısının çok fazla olmasından dolayı yöntemin iyi ölçeklenebilir olmamasıdır.

Foo ve arkadaşları Ke ve arkadaşlarının yöntemini temel alarak önerdikleri yeni yöntemde [1] görüntülerden elde edilen SIFT öznitelikleri bir budamadan geçirilerek dizinlenmiş ve bu şekilde dizinlenen SIFT özniteliklerinin sayısı sınırlandırılmıştır. Bu şekilde ölçeklenebilirlik kısıtını aşmayı planlamışlardır. Bu yöntemde sadece önceden belirlenmiş sayıda öznitelik, görüntü üzerinde buldukları noktaların ışık seviyelerine göre çoktan aza doğru sıralanarak seçilmiş ve dizinlenmiştir. Bu şekilde doğruluktaki küçük bir kayba karşın sorgu zamanı oldukça azaltılmış ve dolayısıyla ölçeklenebilirlik artırılmıştır. Bu yöntemle elde edilen sonuçlara baktığında yöntemin ışık seviyesi ve renk dengesi değişimlerine karşı çok etkili olamadığı görülmektedir.

Bir diğer öznitelik tabanlı neredeyse özdeş görüntü erişimi yöntemi Zhao ve arka-

daşları tarafından geliştirilmiştir [17]. Geliştirdikleri bu yöntemde neredeyse özdeş doğrulaması ölçek ve döndürmeden bağımsız özniteliklerin desen entropisi kullanılarak yapılmıştır. Ölçek ve döndürmeden bağımsız özniteliklerin desen entropisi her bir öznitelik çifti arasında çizilen doğrunun yatayla yaptığı açının histogramının çıkarılarak bu açıların uyumunun kontrol edilmesidir.

## 2.4 İki Öznitelik Kümesi Arasındaki Benzerliği Ölçmek İçin Kullanılan Yöntemler

Bu bölümde literatürde iki öznitelik kümesi arasındaki benzerliği hesaplamak için kullanılan uzaklık metrikleri ele alınacaktır. Bu metrikler genel olarak üç alt gruba ayrılabilirler. Bunlardan birincisi noktalar arası (bin-to-bin) uzaklık hesaplayan metriklerdir ve örnek olarak Minkowski-Form uzaklık ailesi olarak bilinen  $L_p$  normları ve  $X^2$  metrikleri noktalar arası uzaklık hesaplarlar. İkinci grup ise çapraz noktalar arası (cross-bin) uzaklık hesaplayan metriklerdir. Bu gruba örnek olarak ise Toprak Taşıyıcı uzaklığı gösterilebilir. Üçüncü grup ise bu iki sınıfa dahil edilemeyen uzaklık metrikleridir ve örnek olarak Hausdorff uzaklığı bu uzaklık grubuna dahil edilebilir. Aşağıdaki bu metriklerin kısa özetleri verilmiştir.

### 2.4.1 Minkowski-form uzaklık ailesi

Minkowski-Form uzaklık ailesi aynı zamanda  $L_p$  normları olarak da bilinir.  $P$  ve  $Q$  iki farklı aynı eleman sayısına sahip noktalar kümesi olmak üzere  $L_p$ 'nin genel hesaplama formülü aşağıda verilmiştir.

$$L_p(P, Q) = \left( \sum_{i=1}^n (|p_i - q_i|)^p \right)^{\frac{1}{p}}, \quad (2.1)$$

burada  $p_i$  ve  $q_i$  sırasıyla  $P$  ve  $Q$  içerisindeki  $i$ . noktaları ifade etmektedir.

$p = 1$  için  $L_p$  özel bir hali olan  $L_1$  uzaklığına dönüşür ve bu uzaklık aynı zamanda Manhattan uzaklığı olarak bilinir. Benzer şekilde  $p = 2$  için  $L_p$  öklit uzaklığına ( $L_2$ ) dönüşür.

### 2.4.2 X kare uzaklığı

$X^2$  uzaklığı aynı eleman sayısına sahip iki noktalar kümesi için tanımlanmış bir metriktir [24].  $P$  ve  $Q$  iki farklı aynı eleman sayısına sahip noktalar kümesi olmak üzere  $X^2$  uzaklığı aşağıdaki formül ile hesaplanır.

$$X^2(P, Q) = \frac{1}{2} \sum_{i=1}^n \frac{(p_i - q_i)^2}{(p_i + q_i)} , \quad (2.2)$$

burada  $p_i$  ve  $q_i$  sırasıyla  $P$  ve  $Q$  içerisindeki  $i$ . noktaları ifade etmektedir.

### 2.4.3 Hausdorff uzaklığı

Hausdorff uzaklığı iki noktalar kümesi arasındaki uzaklığı veren bir metriktir.  $P$  ve  $Q$  iki noktalar kümesi olmak üzere Hausdorff uzaklığı aşağıdaki formül ile hesaplanır.

$$d_H(P, Q) = \max(\sup_{p \in P} \inf_{q \in Q} d(p, q), \sup_{q \in Q} \inf_{p \in P} d(p, q)) , \quad (2.3)$$

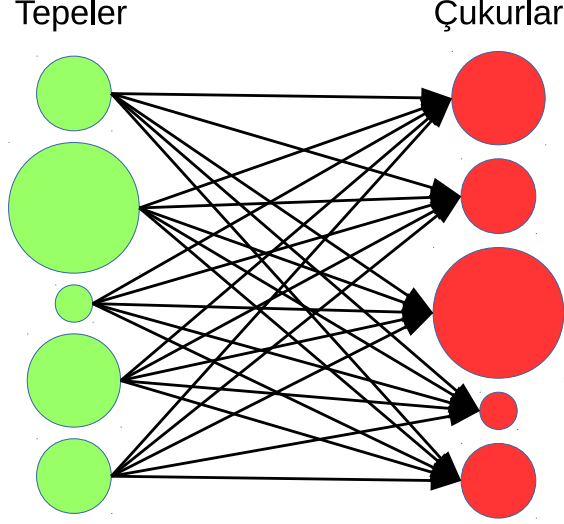
burada  $d(p, q)$ ,  $p$  ile  $q$  noktası arasındaki uzaklığı veren fonksiyonu;  $\sup$  noktalar kümesi içerisindeki bir alt kümenin maksimumunu ve  $\inf$  ise noktalar kümesi içerisindeki bir alt kümenin minimumunu ifade etmektedir.

Literatürde Hausdorff uzaklığının çok daha farklı hesaplama yöntemleri ve bununla birlikte bir çok kullanım alanı vardır [25, 26, 27, 28, 29]. Hausdorff ayrıca iki öznitelik kümesi arasındaki uzaklığı ölçmek için de kullanılabilir. İki öznitelik kümesinin benzerliğini ölçmek için Hausdorff uzaklığı birinci öznitelik kümesindeki minimum ile ikincideki maksimum arasındaki uzaklık ile birincideki maksimum ile ikincideki minimum arasındaki uzaklığın maksimumunu alınarak hesaplanabilir.

### 2.4.4 Toprak taşıyıcı uzaklığı

Toprak Taşıyıcı Uzaklığı (EMD [30, 31, 32, 33]) iki dağılımdan birinin bir diğerine en az masrafla nasıl dönüştürülebileceğini hesaplar. Şekil 2.1'de EMD algoritması görsel olarak ifade edilmiştir. Şekilde görüleceği üzere bir tarafta farklı büyüklüklerde tepeler diğer tarafta ise yine farklı büyüklüklerde çukurlar vardır. Oklar ise tepelerden çukurlara olan yolları simgelemektedir. Tepelerdeki fazlalıklar ile çukurları en az masrafla doldurmak için hesaplanan değere EMD uzaklığı denir.





Şekil 2.1: Tepeler ve çukurlardan oluşan örnek bir dağılım

Burada masraf her bir tepeden çukura taşınan toprak ağırlığı ile yol uzunluklarının çarpımlarının toplamına eşittir. EMD uzaklığı ne kadar azsa iki dağılım birbirine o kadar çok benzerdir.

EMD algoritmasının daha matematiksel ifadesi şu şekildedir. Öncelikle  $P$  ve  $Q$  olarak aralarında EMD uzaklığı hesaplanacak olan iki dağılım olsun ve  $P = \{(p_1, w_{p_1}), (p_2, w_{p_2}), \dots, (p_n, w_{p_n})\}$  ve  $Q = \{(q_1, w_{q_1}), (q_2, w_{q_2}), \dots, (q_m, w_{q_m})\}$  şeklinde tanımlansınlar. Burada  $p$ 'ler ve  $q$ 'lar dağılımlar içindeki noktaları ve  $w$ 'lar ise noktaların ağırlıklarını ifade etmektedir.  $D = [d_{i,j}]$  ise  $p_i$  noktaları ile  $q_j$  noktaları arasındaki uzaklıklar olsun.

EMD uzaklığı  $P$  ile  $Q$  arasındaki toplam maliyeti minimize eden bir  $F = [f_{i,j}]$  akışıdır. Burada  $f_{i,j}$ ,  $p_i$  ile  $q_j$  arasındaki akıştır. Bu minimum maliyetin matematiksel ifadesi aşağıdaki formülde verilmiştir.

$$C_{min} = \min \sum_{i=1}^n \sum_{j=1}^m f_{i,j} d_{i,j} , \quad (2.4)$$

burada  $C_{min}$  minimum maliyeti ifade etmektedir ve bu formül için şu kısıtlar dikkate alınmalıdır:

$$f_{i,j} \geq 0, 1 \leq i \leq n, 1 \leq j \leq m$$

$$\sum_{j=1}^m f_{i,j} \leq w_{p_i}, 1 \leq i \leq n$$

$$\sum_{i=1}^n f_{i,j} \leq w_{q_j}, 1 \leq j \leq m$$

$P$  ile  $Q$  arasındaki toplam akış ise aşağıdaki formül ile hesaplanır.

$$\sum_{i=1}^n \sum_{j=1}^m f_{i,j} = \min\left(\sum_{i=1}^n w_{p_i}, \sum_{j=1}^m w_{q_j}\right) \quad (2.5)$$

Sonuç olarak maliyeti minimum yapacak ideal  $F$  akışı yukarıdaki doğrusal eniyileme problemini çözerek bulunabilir.  $P$  ve  $Q$  arasındaki EMD uzaklığı ise toplam maliyetin toplam akış ile normalize edilmesiyle hesaplanır ve bu hesap için gerekli olan formül aşağıda verilmiştir.

$$EMD(P, Q) = \frac{\min \sum_{i=1}^n \sum_{j=1}^m f_{i,j} d_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m f_{i,j}} . \quad (2.6)$$



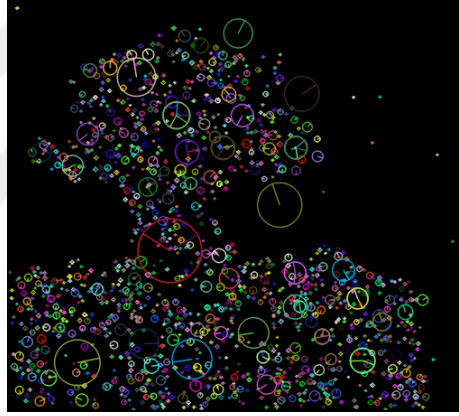
### 3. YOĞUNLUK HARİTASI İLE ÖZNETELİK SEÇİMİ

Daha önceden bahsedildiği üzere neredeyse özdeş görüntüler diğer ortak bölümler içeren görüntülerden daha fazla ortak özneteliğe sahiptir. Bu şekilde çok sayıda ortak özneteliğe sahip olan görüntülerde eşleştirme işlemi için tüm özneteliklerin nitelikli bir alt kümesinin yeterli olacağı tezinden yola çıkarak Foo ve arkadaşları yaptığı çalışmada [1] neredeyse özdeş görüntü tespitini geliştirdikleri yöntemle oluşturdukları öznetelik alt kümesiyle yapmışlardır ve elde ettikleri sonuçlardan böyle bir alt kümenin neredeyse özdeş tespiti için yeterli olduğu anlaşılmıştır. Fakat Foo ve arkadaşlarının çalışmasında özneteliklerin alt kümesi görüntü üzerinde buldukları noktaların ışık değerine göre oluşturulduğu için iyi sonuçlar ancak yeteri kadar çok eleman sayısına sahip bir alt küme ile elde edilebilmiştir. Buna ek olarak görüntü üzerindeki ışık ve renk değişimi seçilen öznetelik alt kümesini doğrudan etkilediği için bu yöntem ışık ve renk değişimlerine yeteri kadar dirençli değildir.

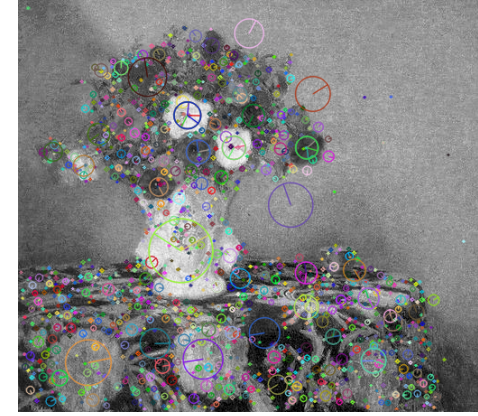
Temel olarak neredeyse özdeş tespiti için oluşturulacak öznetelik alt kümesinin nitelikli olması gerekmektedir. Burada nitelikli olmasından kasıt, bir görüntüde seçilen özneteliklerin bu görüntünün neredeyse özdeşlerinde seçilen özneteliklerle olabildiğince fazla örtüşmesidir. Bu tezde bu örtüşmeyi maksimize etmek için geliştirilen yöntemde görüntü üzerindeki değişimlere dirençli bölgelerin belirlenerek özneteliklerin bu dirençli bölgelerdeki kuvvetli özneteliklerden seçilmesi sağlanmıştır. Bu yöntemde öznetelikler yine öznetelikler ile oluşturulan yoğunluk haritası kullanılarak seçilmiştir. Çünkü oluşturulan yoğunluk haritası ışık ve renk değişimleriyle kolay değişmez ve bu değişmezlik ile seçilecek öznetelikler diğer neredeyse özdeşlerdeki seçilenlerle daha çok örtüşmüş olur. Bir başka deyişle geliştirilen yöntemle oluşturulan öznetelik alt kümesi neredeyse özdeş tespiti için daha nitelikli bir alt küme olacaktır. Bu ise daha küçük bir öznetelik alt kümesi ile daha iyi sonuçlar elde edilmesini demektir. Geliştirilen yöntem temel olarak şu üç aşamadan oluşmaktadır: öznetelik çıkarma, yoğunluk haritasının oluşturulması ve son olarak öznetelik seçimi (bkz. Şekil 1.1). Şimdi bu aşamaları daha ayrıntılı bir şekilde açıklayalım.



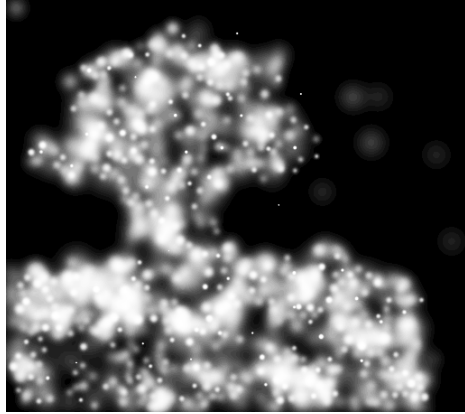
(a)



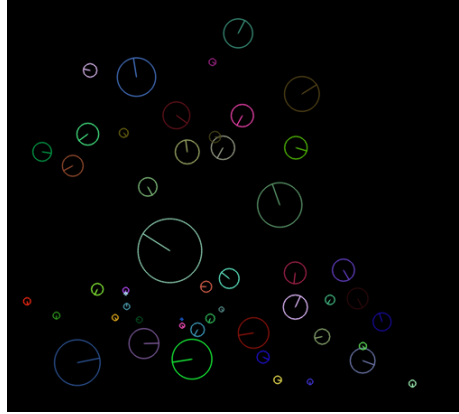
(b)



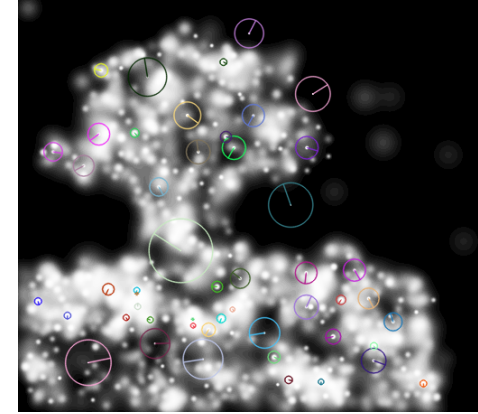
(c)



(d)

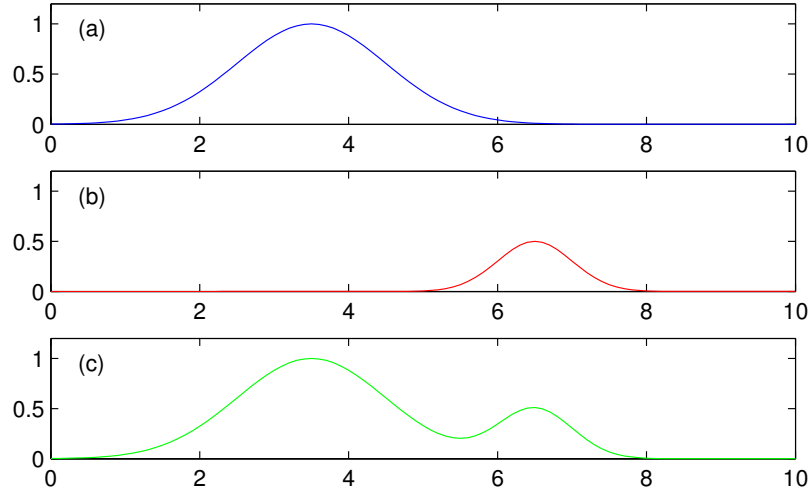


(e)



(f)

Şekil 3.1: Örnek bir görüntü için öznitelik seçimi (a) Örnek görüntü (b) Tüm SIFT öznitelikleri (c) Tüm özniteliklerle birlikte örnek görüntü (d) Hesaplanan yoğunluk haritası (e) Seçilen öznitelikler (f) Seçilen özniteliklerle birlikte örnek görüntü



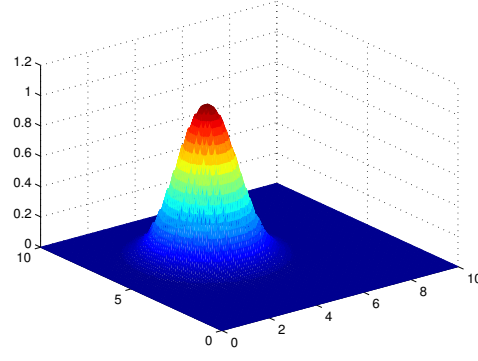
Şekil 3.2: Doğru üzerindeki 2 öznitelik noktası için yoğunluk haritası oluşturma (a) 3.5 noktasında bulunan birincinin Gaussian katkısı (b) 6.5 noktasında bulunan ikincinin Gaussian katkısı, (c) edilen yoğunluk haritası

### 3.1 Öznitelik Çıkarma

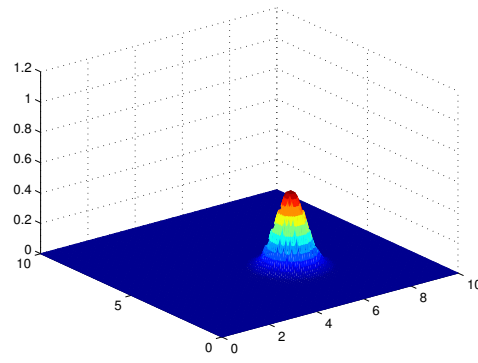
Bu aşamada görüntü üzerindeki tüm SIFT öznitelikleri ve bu özniteliklerin tanımlayıcıları hiç bir filtrelemeye tabi tutulmadan çıkarılmıştır. Bu çalışmada biz SIFT algoritması olarak Lowe'nin orijinal Gaussian Farkları (DoG [18]) algılayıcısı kullanılmıştır. Öznitelik tanımlayıcısı olarak ise yine orijinal SIFT tanımlayıcısı kullanılmıştır. Şekil 3.1b'de Şekil 3.1a'da verilen örnek görüntüden elde edilen toplam 1676 SIFT özniteliği verilmiştir. Bu özniteliklerin görüntü üzerindeki gösterimi ise Şekil 3.1c'de verilmiştir.

### 3.2 Yoğunluk Haritası Oluşturma

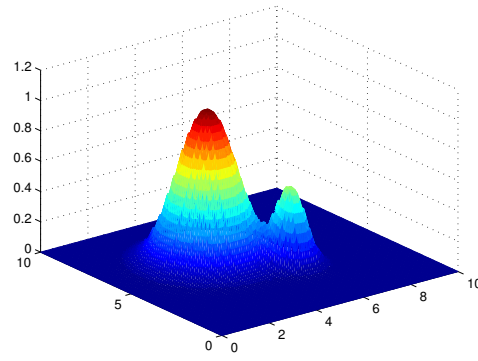
Görüntü üzerindeki tüm SIFT öznitelikleri bulunduktan sonra artık yoğunluk haritası oluşturma işlemine geçilir. Walker ve arkadaşları [34] belirgin yüzsel öznitelikleri bulmak için bir yoğunluk haritası tahmin yöntemi kullanmışlardır. Bu yöntemde göre görüntü üzerindeki her bir noktanın yoğunluğu, görüntü üzerindeki özniteliklerin o noktaya Gaussian katkısının toplamı şeklinde hesaplanmıştır. Daha sonra belirgin yüzsel öznitelikler oluşan haritanın daha az yoğun bölgelerinden başlanarak seçmiştir.



(a)



(b)



(c)

Şekil 3.3: Bir görüntü üzerindeki 2 öznitelik noktası için yoğunluk haritası oluşturma (a) 1. özniteliğin Gauss katkısı (b) 2. özniteliğin Gauss katkısı (c) Yoğunluk haritasının 1. ve 2. özniteliklerin Gauss katkılarının toplamı ile oluşmuş hali

Bizim yoğunluk haritası oluşturmak için kullandığımız yöntem de Walker ve arkadaşlarının kullandığı yöntemle bir kaç önemli farklılık dışında benzerdir. Birincisi biz yoğunluk haritası için görüntünün diferansiyel yapısından hesaplanan sabit değerleri değil SIFT özniteliklerini kullandık. İkincisi ve daha önemlisi öznitelik-

ler daha az yoğun bölgelerin aksine daha yoğun bölgelerden başlanarak seçildi. Bu şekilde tekrar olasılığı yüksek özniteliklerin seçimi hedeflendi.

Görüntü üzerinde koordinatları  $(x,y)$  olan bir noktanın yoğunluğunu hesaplamak için geliştirilen formül aşağıda verilmiştir.

$$\hat{p}(x,y) = \sum_{i=1}^N S_i \varphi_{\mu,\sigma^2}(\sqrt{(x-x_i)^2 + (y-y_i)^2}) , \quad (3.1)$$

burada  $N$  öznitelik sayısını,  $S_i$  öznitelik  $i$ 'nin ölçeğini,  $x_i$  ve  $y_i$  onun konumunu,  $\mu$  Gauss beklenen değerini ve  $\sigma^2$  Gauss varyansını ifade etmektedir. Bu formülde sırasıyla  $\mu$  ve  $\sigma^2$  0 ve  $1/S_i$ 'ye eşittir. Formüle göre bir noktanın yoğunluğu her bir öznitelğin o noktaya olan Gaussian katkıları toplanarak elde edilir.

Formülde bir öznitelğin yoğunluk haritasına ne kadar katkı sağlayacağını  $S_i$  yani öznitelğin ölçeği belirlemektedir. Bir öznitelğin ölçeği SIFT yöntemine göre oluşturulan görüntü piramidinde öznitelğin kaçınıcı katmanda bulunduğudur. Bu aynı zamanda bir öznitelğin etkin bölgesini de göstermektedir. Yapılan gözlemlerde ölçeği büyük olan özniteliklerin neredeyse özdeşlerde bulunma olasılığının yüksek olduğu görülmüştür. Bu nedenle öznitelğin yoğunluk haritasına katkısı öznitelğin ölçeğiyle orantılı olarak belirlenmiştir.

Bir doğru üzerindeki 2 öznitelik noktası için yoğunluk haritasının nasıl elde edildiği Şekil 3.2'de verilmiştir. Şekilde bir noktanın yoğunluğunun iki öznitelikten gelen yoğunluk toplamlarından elde edildiği görülmektedir. Bir düzlem üzerindeki Gaussian katkısı ve yine iki öznitelğin bulunduğu bir düzlemin yoğunluk haritası ise Şekil 3.3'de verilmiştir.

$n \times m$  boyutlarında ve  $N$  tane SIFT öznitelği çıkarılan bir görüntü için yoğunluk haritası öznitelik noktalarının yoğunluklarının bir döngü ile hesaplanması sonucu elde edilir. Şekil 3.1d'de örnek görüntü için görüntü üzerindeki her bir noktanın yoğunluğu hesaplanarak elde edilen yoğunluk haritası verilmiştir. Öznitelik seçimi sadece öznitelik noktaların yoğunluklarına bakılarak elde edildiği için görüntü üzerindeki her bir noktanın yoğunluğu yerine sadece öznitelik noktalarının yoğunluklarının hesaplanması yeterli olacaktır. Bu şekilde Walker ve arkadaşlarının yönteminden [34] farklı olarak yoğunluk hesabı  $\mathcal{O}(nmN)$  karmaşıklıktan  $\mathcal{O}(N^2)$  karmaşıklığa indirilmiş olacaktır.

### 3.3 Öznitelik Seçimi

Öznitelik noktalarının yoğunluk haritası hesaplandıktan sonra daha önceden de belirtildiği üzere öznitelikler çok yoğun bölgelerden başlanarak seçilmiştir. Çünkü çok yoğun bölgelerin değişime daha dirençli bölgeler olduğu öngörülmüştür. Fakat burada dikkat edilmesi gereken noktalardan bir tanesi bir yoğun bölgeden kaç tane özneliğin seçileceği konusudur. Yani eğer seçilen tüm öznitelikler doğrudan görüntünün en yoğun bölgesinden seçilirse bu defa da değişime dayanıksız özniteliklerden de seçilmiş olacaktır. Bu problemin önüne geçmek için ise, bir öznitelik seçildikten sonra onun çevresindeki özniteliklerden tekrar seçim yapılmamıştır.

Özniteliklerin yoğunluk haritası kullanılarak seçiminin ve dolayısıyla bir alt kümesinin oluşturulması Algoritma 1'de görüldüğü gibi yapılmaktadır. Bu algoritma bir görüntüyü ve seçilecek öznitelik sayısını girdi olarak alır ve çıktı olarak görüntüden elde edilen SIFT özniteliklerinin girdide verilen öznitelik sayısı kadar elemanı olan alt kümesini verir. Algoritma, çalışmaya görüntünün SIFT özniteliklerini çıkarmakla başlar. Daha sonra yoğunluk haritası hesaplanır. Yoğunluk haritasına göre en yoğun noktada bulunan öznitelik seçilir ve sonuç listesine eklenir. Daha sonra farklı yoğun bölgelerden de seçim yapılabilmesi amacıyla bu öznitelik ve etrafındaki öznitelikler seçim yapılacak listeden çıkarılırlar (burada özneliğin ne kadarlık etrafındaki özniteliklerin listeden kaldırılacağı ampirik olarak belirlenmiştir). Daha sonra bu işlem listede kalan en yoğun noktadaki öznitelik için tekrarlanır ve bu işleme istenilen sayıda öznitelikten oluşan küme elde edilene kadar devam edilir. Şekil 3.1a'de verilen örnek görüntü için bu algoritma ile seçilmiş öznitelikler ve bu özniteliklerin yoğunluk haritası üzerindeki gösterimi Şekil 3.1e ve Şekil 3.1f'de verilmiştir.

---

#### Algorithm 1 Öznitelik Seçme

---

**Require:**  $G$  görüntüsü,  $K$  seçilecek öznitelik sayısı

**Ensure:**  $R$  seçilen öznitelik listesi

- 1:  $G$  görüntüsünün SIFT özniteliklerini çıkar ve  $L$  listesine ekle
  - 2: özniteliklerden yoğunluk haritası oluştur
  - 3: boş bir  $R$  listesi oluştur
  - 4: **for**  $i = 1$  to  $K$  **do**
  - 5:   en yoğun noktadaki özneliği  $R$  listesine ekle
  - 6:   bu öznitelik ve yakınındaki öznitelikleri  $L$  listesinden sil
  - 7: **end for**
  - 8: **return**  $R$
-



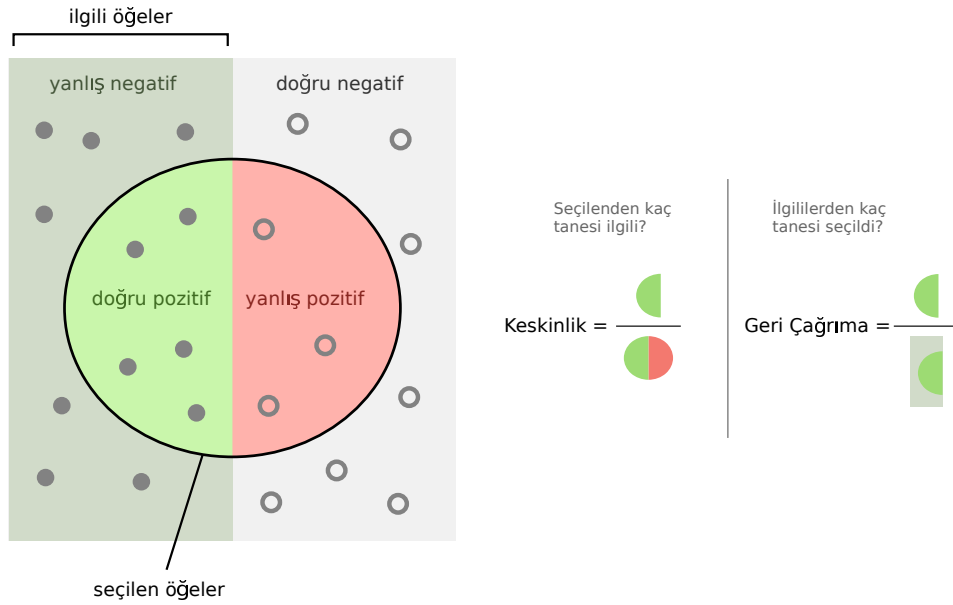
## 4. DENEYSEL ÇALIŞMALAR

Bu bölümde geliştirilen yöntemin deneysel olarak nasıl test edildiğinden bahsedilmiştir ve elde edilen sonuçlar bu bölümde paylaşılmıştır.

### 4.1 Başarım Metrikleri

Deney sonuçlarının değerlendirilebilmesi için bazı başarımler metriklerine ihtiyaç duyulmaktadır. Yapılan testlerin çıktıları göz önünde bulundurulduğunda bu testlere en uygun başarımler metriklerinin geri çağırma, keskinlik ve bu ikisinden hesaplanan f-değeri olduğu görülmektedir. Şimdi sırasıyla bu 3 başarımler metriğine kısaca bakalım.

**Geri Çağırma:** Kısaca ilgili öğelerden kaç tanesi seçildi sorusunun cevabıdır. Bu metrik veri kümesi içerisinde verilen sorgu görüntüsünün neredeyse özdeşlerinin kaçta kaçının yöntem tarafından seçilebildiğidir. Bu tezde geri çağırma  $R$  ile ifade edilecektir.



Şekil 4.1: Keskinlik ve Geri Çağırma'nın görsel olarak ifadesi. (Kaynak: İngilizce Wikipedia sayfasından alınarak düzenlenmiş ve türkçeleştirilmiştir)

**Keskinlik:** Bu metrik yöntem tarafından veri kümesi içerisinde sorgu görüntüsünün neredeyse özdeşi olarak seçilen görüntülerin gerçekte kaçta kaçının sorgu görüntüsünün neredeyse özdeşi olduğudur. Bu tezde keskinlik  $P$  ile ifade edilecektir.

**F-değeri:** Bu metrik ise geri çağırma ve keskinlik metriklerini bir metrikte birleştirir ve yöntemlerin bir değer üzerinde karşılaştırılmasını sağlar. Bu metrik  $F$  ile ifade edilecektir. F-değeri aşağıdaki formül ile hesaplanmaktadır.

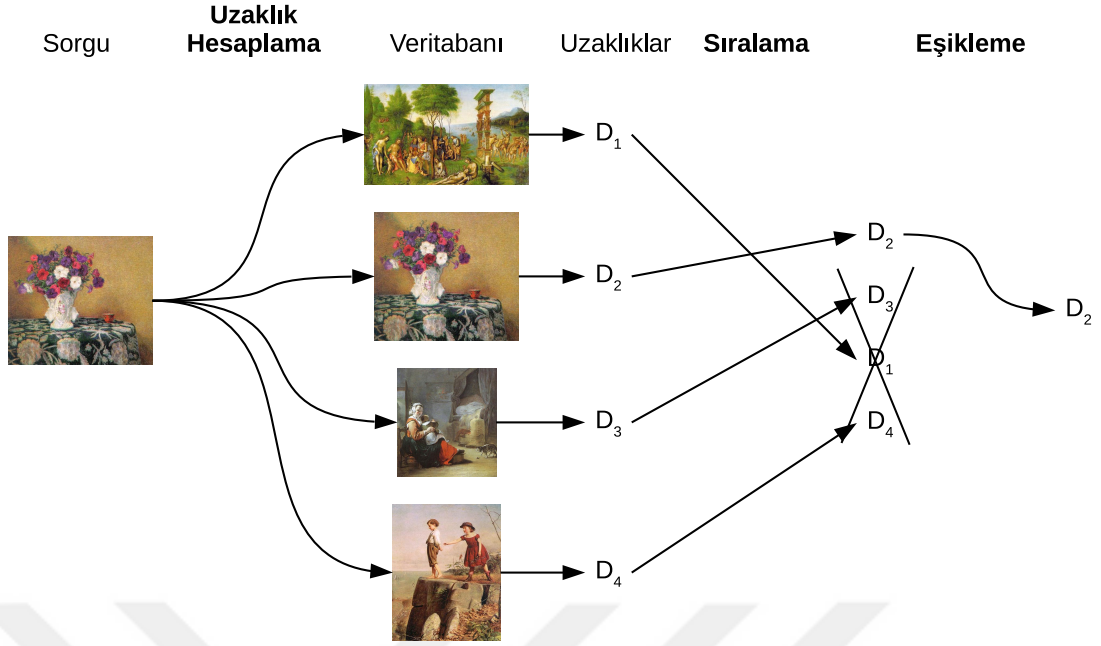
$$F = 2 \frac{P \times R}{P + R} , \quad (4.1)$$

## 4.2 Değerlendirme Yöntemi

Yapılan deneyler iki adımdan oluşmaktadır. Birinci adım ön işlem adımdır ve bu adımda veri kümesindeki bütün resimlerden elde edilen seçilmiş SIFT öznitelikleri bu özniteliklerin tanımlayıcıları ile birlikte bir dosyaya kaydedilir. Ön işlem adımının algoritması Algoritma 2’de verilmiştir. Bu algoritmaya göre öncelikle boş bir dosya oluşturulur. Daha sonra görüntü kümesindeki her bir görüntüden öznitelik seçimi yapılır ve bu seçim sonucu elde edilen öznitelik kümesi ve bu kümenin tanımlayıcıları dosyaya eklenir. Son olarak ön işlem adımı oluşan dosyanın kaydedilmesi ile sonlandırılır. Ön işlem adımı veri kümesi verildikten sonra çevrimdışı olarak yapılır. Bu nedenle ön işlemin uzun sürmesi sorgulama zamanını etkilemez.

İkinci adım verilen bir sorgu görüntüsü için veri kümesinden bu görüntüsünün neredeyse özdeşlerinin bulunması aşaması olan sorgulama adımdır. Bu adımda öncelikli olarak iki görüntü arasındaki uzaklığı hesaplayacak bir algoritmaya ihtiyaç vardır.

Bu tezde, iki görüntü arasındaki uzaklık, görüntülerden seçilen özniteliklerle FastEMD algoritmasıyla [33] yardımıyla hesaplanmıştır. Bu algoritma ile noktalar arasındaki mesafelere belirli bir eşik değeri konularak eşik değeri üzerindeki yolların mesafesinin eşik değeri kadar olması sağlanmıştır ve bu şekilde eşik değerinden küçük mesafeler için ağırlık taşınması normal EMD algoritması ile yapılırken geriye kalan ağırlık doğrudan eşik değeri ile çarpılarak sonuca eklenmiştir. Bu şekilde duruma göre çok daha hızlı bir EMD hesabı yapılır. EMD hesabında özniteliklerin uzaydaki konumu o özniteliğin SIFT tanımlayıcısı ile ifade edilmiştir ve iki öznitelik arasındaki yol uzaklığı bu iki özniteliğin tanımlayıcıları arasındaki öklit



Şekil 4.2: Bir sorgu görüntüsü için neredeyse özdeşlerin veritabanından bulunma yöntemini anlatan şekil. Bu şekilde koyu renkli yazılmış başlıklar yapılan işlemleri ifade etmektedir. Açık renkliler ise resimdeki nesnelerin adları için kullanılmıştır.

uzaklığı olarak alınmıştır. Özniteliklerin ölçek değerleri ise o özneliğin uzaydaki konumunun ağırlığı olarak alınmıştır. EMD doğası gereği yavaş çalışan bir algoritmadır fakat hata toleransının çok iyi olmasından dolayı bu algoritma tercih edilmiştir.

Sorgulama adımı Şekil 4.2’de görsel olarak özetlenmiştir. Şekilde görüleceği üzere verilen bir sorgu görüntüsü için bu görüntü ile veri kümesindeki her bir görüntü arasındaki uzaklık EMD ile hesaplanır. Ardından bu uzaklıklar küçükten büyüğe sıralanır ve önceden belirlenmiş bir eşik değeri ile sıralanan uzaklıklar eşiklenir. Yani sorgu görüntüsü ile arasındaki uzaklığı eşik değerinden küçük olan resimler sorgu görüntüsünün neredeyse özdeşi olarak işaretlenirler.

Algoritma 3’te sorgulama adımının nasıl gerçekleştirildiği verilmiştir. Bu algoritma veri kümesinden elde edilen özniteliklerin kaydedildiği dosyayı, sorgu görüntüsünü ve neredeyse özdeş olup olmadığı kararını vermede gerekecek olan eşik değerini girdi olarak alır, veri kümesindeki her bir görüntü ile sorgu görüntüsü arasındaki uzaklığı hesaplar ve çıktı olarak verilen sorgu görüntüsünün neredeyse özdeşlerinin listesini verir. Algoritma çalışmaya ön işlem aşamasında elde edilen ve veri kümesindeki görüntülerin seçilmiş özniteliklerininin kayıtlı olduğu dosyayı belleğe okumayla başlar. Dosya belleğe aktarıldıktan sonra sorgu görüntüsünden Algoritma 1 ile öznitelikleri seçer. Neredeyse özdeşlerin listesinin tutmak için boş bir liste oluşturur. Daha sonra veri kümesindeki her bir görüntünün öznitelikleri

ile sorgu görüntüsünün öznitelikleri arasındaki EMD uzaklığını hesaplar ve hesaplanan bu uzaklık eşik değerinden küçük ise veri kümesinden bu özniteliklere karşılık gelen görüntüyü sonuç listesine ekler. Son olarak oluşan sonuç listesini çıktı olarak verir.

---

**Algorithm 2** Ön İşlem

---

- 1: boş bir  $D$  dosyası oluştur
  - 2: **for** veri kümesindeki her bir görüntü için **do**
  - 3:   o görüntüden Algoritma 1 ile seçilen öznitelikleri  $L$  listesine kaydet
  - 4:    $L$  listesini  $D$  dosyaya ekle
  - 5: **end for**
  - 6:  $D$  dosyasını kaydet
- 

---

**Algorithm 3** Sorgulama

---

**Require:**  $D$  öznitelik dosyası,  $S$  sorgu görüntüsü, ve  $E$  eşik değeri

**Ensure:**  $L$  neredeyse özdeş listesi

- 1:  $D$  dosyasındaki öznitelikleri belleğe oku
  - 2:  $S$  görüntüsünden Algoritma 1 ile öznitelikleri seç ve  $P$  kümesine kaydet
  - 3: boş bir  $L$  listesi oluştur
  - 4: **for** veri kümesindeki her bir  $G$  görüntüsü için **do**
  - 5:   bellekten  $G$  görüntüsünün öznitelik kümesini ( $R$ ) al
  - 6:    $P$  ile  $R$  öznitelik kümesi arasındaki EMD uzaklığını hesapla
  - 7:   **if** uzaklık  $E$ 'den küçükse **then**
  - 8:     veri kümesinden  $P$ 'ye karşılık gelen görüntüyü  $L$ 'ye ekle
  - 9:   **end if**
  - 10: **end for**
  - 11: **return**  $L$
- 

### 4.3 Veri Kümesi

Önerilen yöntemin testi için Ke ve arkadaşlarının kullandığı [35] çevrimiçi bir resim galerisinden elde edilen resimler kullanılmıştır. Bu galeriden elde edilen resimlerden 3 farklı veri kümesi oluşturulmuştur. Bu veri kümeleri sırasıyla rastgele seçilmiş 5, 100 ve 500 resimden oluşmaktadır. Veri kümelerinden örnek resimler Şekil 4.1'de verilmiştir.



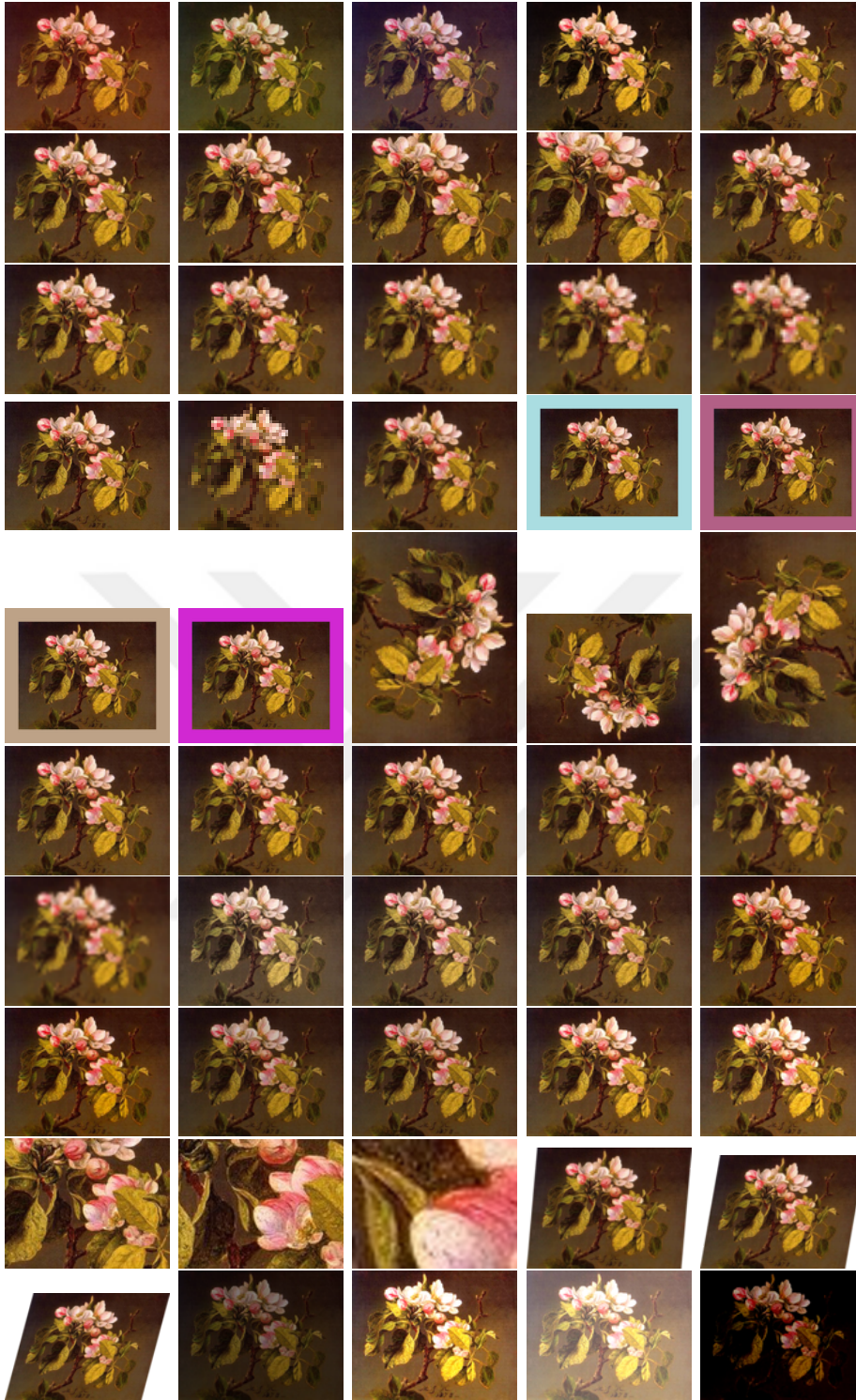
Resim 4.1: Oluşturulan veri kümesinden 24 örnek görüntü

Daha sonra her bir veri kümesindeki her bir orijinal resim için [15]'de tarif edildiği şekilde 50 farklı yaygın kullanılan yöntemden neredeyse özdeş resimler oluşturulmuş ve toplamda 255, 5100 ve 25500 resimden oluşan 3 veri kümesi elde edilmiştir. Aşağıda bu 50 yaygın kullanılan yöntem ve parantez içerisinde her bir yöntemden kaç adet neredeyse özdeş görüntü oluşturulduğu listelenmiştir.

- Görüntüdeki her bir noktanın kırmızı renginin %10 artırılması ile elde edilen görüntü (1)
- Görüntüdeki her bir noktanın yeşil renginin %10 artırılması ile elde edilen görüntü (1)
- Görüntüdeki her bir noktanın mavi renginin %10 artırılması ile elde edilen görüntü (1)
- Görüntüdeki kontrast farkının %10 artırılması ile elde edilen görüntü (1)
- Görüntüdeki kontrast farkının %10 azaltılması ile elde edilen görüntü (1)
- Görüntünün kenarlardan başlayarak sırasıyla %5, %10, %20 ve %30 oranında kırılmasıyla elde edilen görüntüler (4)
- Görüntüdeki siyah noktaların giderilmesi ile elde edilen görüntü (despeckle işlemi) (1)
- Görüntünün örneklenerek yeni boyutun orijinalinin sırasıyla %90'ı, %80'i, %70'i, %60'ı, %50'si, %30'u ve %10'u olacak şekilde görüntüler elde edilmesi (7)
- Görüntü saklama formatının jpeg'den gif'e dönüştürülmesi ile elde edilen görüntü (1)
- Görüntüye 4 farklı rastgele renkte boyutunun %10'u çerçeve eklenerek elde edilen görüntüler (4)
- Görüntünün sırasıyla 90°, 180° ve 270° döndürülmesiyle elde edilen görüntüler (3)
- Görüntünün sırasıyla 2, 4, 8 kat büyütülüp sonra tekrar eski boyutuna getirilmesiyle elde edilen görüntüler (3)
- Görüntünün sırasıyla 2, 4, 8 kat küçültülüp sonra tekrar eski boyutuna getirilmesiyle elde edilen görüntüler (3)

- Görüntüdeki her bir noktanın doygunluğunun sırasıyla %70, %80, %90, %110 ve %120 oranında değiştirilmesiyle elde edilen görüntüler (5)
- Görüntüdeki her bir noktanın ışık değerinin sırasıyla %80, %90, %110 ve %120 oranında değiştirilmesiyle elde edilen görüntüler (4)
- Görüntünün kenarlardan başlayarak sırasıyla %50, %70 ve %90 oranında kırılmasıyla elde edilen görüntüler (3)
- Görüntüye sırasıyla saat yönünde  $5^\circ$ ,  $10^\circ$  ve  $15^\circ$  burulma uygulayarak elde edilen görüntüler (3)
- Görüntüdeki her bir noktanın ışık değerinin sırasıyla %50 ve %150 oranında değiştirilmesiyle elde edilen görüntüler (2)
- Görüntüdeki her bir noktanın kontrastının 3 kat artırılması ve 3 kat artırılmasıyla elde edilen görüntüler (2)

Görüntüler üzerindeki bütün bu dönüşümler ImageMagick [36] programı kullanılarak yapılmıştır. Bu 50 farklı dönüşümün veri kümesindeki örnek bir resme uygulanması sonucu elde edilen yeni görüntüler soldan sağa ve yukarıdan aşağıya üstteki listeye uygun sırada Şekil 4.2'de görülmektedir.



Resim 4.2: Örnek bir görüntüye Sayfa 22’de verilen listedeki dönüşümler uygulanarak elde edilen neredeyse özdeş görüntüler soldan sağa ve yukarıdan aşağıya listedeki sırada verilmiştir



#### 4.4 Deney Düzenegi

Bu bölümde önerilen yöntemle karşılaştırılacak yöntemler ve önerilen yöntemin performansını göstermek için yapılan testler hakkında bilgiler verilecektir.

Aşağıda önerilen yöntemin karşılaştırıldığı diğer iki yöntem listelenmiştir.

- **Yöntem 1:** Ke ve arkadaşlarının [15] önerdiği yöntemdir. Bu yöntemde görüntüler üzerindeki tüm öznitelikler neredeyse özdeş tespiti için kullanılmaktadır. Bu yöntem büyük veri kümeleri için çok yavaş çalışmasından dolayı sadece ilk testte kullanılmıştır.
- **Yöntem 2:** Foo ve arkadaşlarının [1] önerdiği yöntemdir. Bu yöntemde ise öznitelikler görüntü üzerinde buldukları noktaların ışık değerine göre sıralanmaktadır ve neredeyse özdeş tespiti için bu sıralı özniteliklerden sadece ilk  $N$  tane öznitelik kullanılmaktadır.

Bir tane ilk veri kümesi, dört tane ikinci veri kümesi ve son olarak bir tane de üçüncü veri kümesi üzerinde olmak üzere toplamda 6 farklı test çalıştırılmıştır. Testlerden bazılarında veri setindeki tüm görüntüler aynı zamanda sorgu görüntüsü olarak da kullanılmıştır. Bu şekilde sorguyla veri setinde bulunan görüntüler arasında iki farklı tür değişiklik bulunacaktır. Mesela sorgu için kullanılan görüntüde kaydırma işlemi yapılmış olsun ve veri setindeki bir görüntüde döndürme işlemi yapılmış olsun. Burada sorgudan veri kümesindeki bu görüntüyü elde etmek için sorgu görüntüsüne öncelikle kaydırma işleminin tersi uygulanmalı ve ardından döndürme uygulanmalıdır. Bu şekilde sorgulama işlemi biraz daha zorlaştırılmış olmaktadır.

Yapılan testler ve açıklamaları aşağıda listelenmiştir.

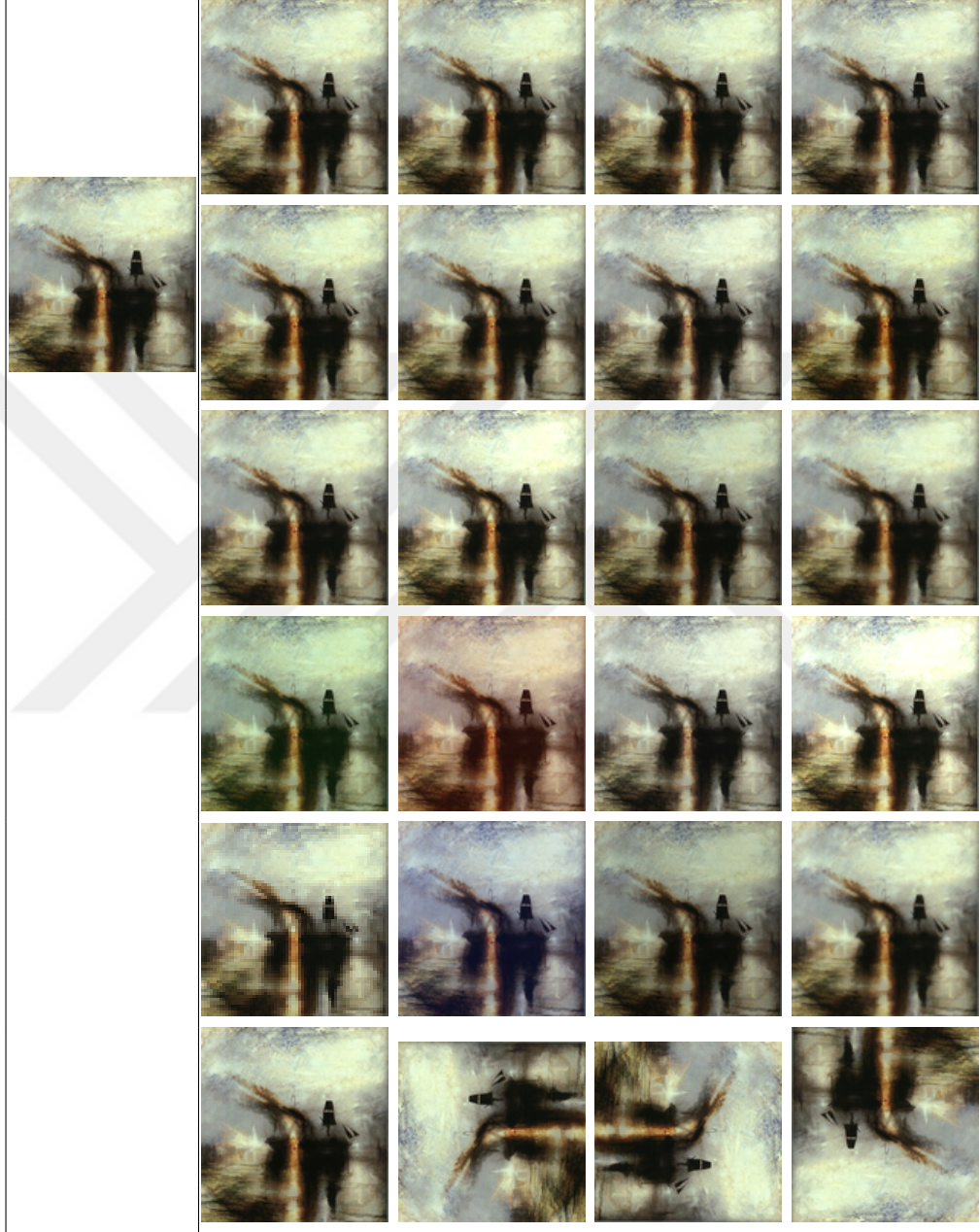
- **Test 1:** Birinci test önerilen ve yukarıda listelenen 2 yöntem olmak üzere 3 neredeyse özdeş tespiti yöntemini karşılaştırmak için içerisinde 255 resim bulunan birinci veri kümesi üzerinde çalıştırılmıştır. Bu ilk testte, bütün resimler sorgu görüntüsü olarak kullanılmıştır ve veri kümesindeki görüntülerin yöntem 2 ve önerilen yöntem için ön işlem adımında 50 şer özniteliği seçilmiştir.
- **Test 2:** Bu testte sorgu için sadece orijinal resimler kullanılmış ve her bir resimden 10 tane öznitelik seçilmiştir. Bu test sadece yöntem 2 ve önerilen yöntem için 5100 resimden oluşan ikinci veri kümesi üzerinde çalıştırılmıştır.

- **Test 3:** Bu test veri kümesindeki her bir resimden 50 şer öznitelik seçimi dışında Test 2 ile aynı şekilde çalıştırılmıştır.
- **Test 4:** Bu test veri kümesindeki her bir resimden 100 er öznitelik seçimi dışında Test 2 ile aynı şekilde çalıştırılmıştır.
- **Test 5:** İkinci veri kümesi üzerinde çalıştırılan bu testte ise resimlerden 50 öznitelik seçilmiş ve tüm resimler sorgu resmi olarak kullanılmıştır.
- **Test 6:** Bu testte ise 25500 resimden oluşan üçüncü veri kümesi kullanılmış, resimlerin her birinden 50 şer öznitelik seçilmiş ve sadece orijinal resimler sorgu için kullanılmıştır.

Tüm bu testler 64 çekirdekli ve 128 GB belleğe sahip bir makine üzerinde GNU parallel [37] aracı ile paralelleştirilerek çalıştırılmıştır. Algoritmalar OpenCV [38] kütüphanesi kullanılarak C++ ve Python dilleri ile yazılmıştır.

#### 4.5 Deneysel Sonuçlar

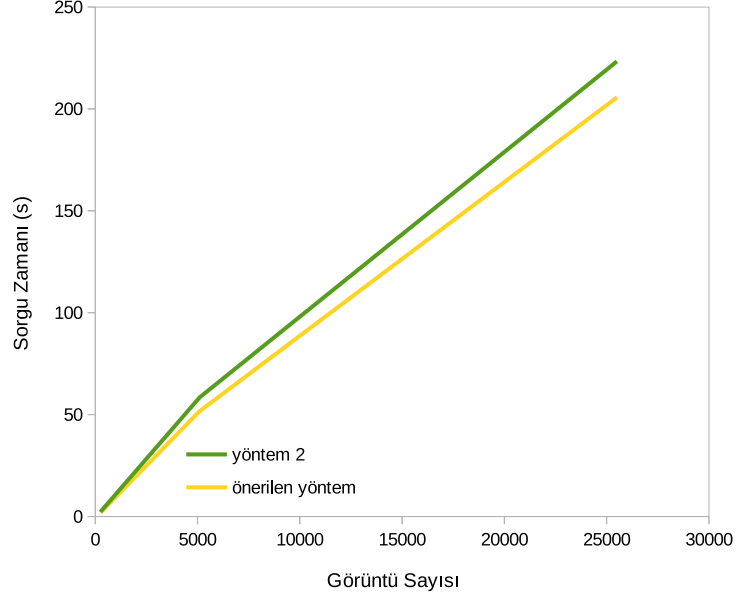
Bu bölümde bir önceki bölümde ayrıntıları verilen 6 deneyin sonuçları tartışılacaktır. Şekil 4.3'de sol tarafta verilen örnek bir sorgu görüntüsü için 5. testin önerilen yöntem için ilk 24 sonucu verilmiştir ve bu 24 görüntünün tamamı sorgu görüntüsünün neredeyse özdeşidir. Bu örnek sorgu görüntüsünden elde edilen toplam SIFT özneliği 883 adettir ve bunlardan sadece 50 tanesi ile neredeyse özdeş tespiti yapılmıştır.



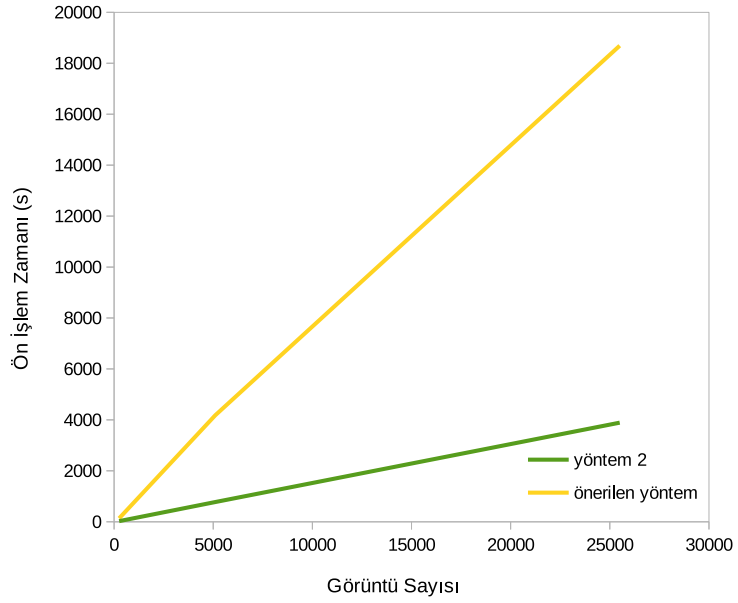
Resim 4.3: Örnek bir görüntü için sorgu görüntüsüyle birlikte sorgu sonucundan sıralı olarak ilk 24 görüntü

Tablo 4.1’de bütün sonuçlar testlere göre verilmiştir. Bu tabloda verilen sonuçlar her bir yöntem için o yöntemin en iyi F-değeri baz alınarak verilmiştir. Sonuçlara bakıldığında önerilen yöntemin F-değerlerinin diğer yöntemlerden daha iyi olduğu görülmektedir. Daha özelden yöntem 1’e göre ilk testte önerilen yöntem 1.5 kat daha iyi bir F-değeri sonucu vermiştir. yöntem 2’ye göre ise F-değerinde yüzde 3’lük bir artış vardır. Elde edilen sonuçlardan dikkat çeken noktalardan bir tanesi tüm öznitelikler kullanılarak sorgulama yapılan yöntem 1’de sonuçların beklendiği gibi iyi çıkmamasıdır. Bunun sebebi EMD algoritmasının bir görüntü için ortalamada 1000’den fazla çıkan öznitelik sayısından kötü etkilenmesidir.

Şekil 4.3a’de önerilen yöntem ve yöntem 2’nin sorgu zamanı-görüntü sayısı grafikleri verilmiştir. Bu grafiklere ve Tablo 4.1’e bakıldığında ise yöntem 1’in sorgu süresinin diğer iki yöntemden çok daha fazla olduğu görülmektedir. Yöntem 2 ile önerilen yöntemin sorgu süreleri kıyaslandığında ise önerilen yöntemin sorgu süresinde yaklaşık yüzde 12’lik bir iyileşme görülmektedir. Sorgu süresindeki bu iyileşmenin sebebi ise kullanılan FastEMD algoritmasının önerilen yöntemde daha iyi sonuçlar vermesinden kaynaklanmaktadır.



(a)



(b)

Şekil 4.3: Görüntü sayısına göre sorgu ve ön işlem zamanı değişimleri (a) Önerilen yöntem ve yöntem 2 için sorgu zamanı-görüntü sayısı grafiği (b) Önerilen yöntem ve yöntem 2 için ön işlem zamanı-görüntü sayısı grafiği

Ön işlem sürelerinde ise yöntem 2'nin diğer yöntemlerden daha iyi olduğu görülmektedir. Önerilen yöntem ve yöntem 2 için ön işlem zamanı-görüntü sayısı grafiği Şekil 4.3b'de verilmiştir. Önerilen yöntemin ön işlem süresinin daha uzun olması büyük çoğunlukla öznitelik seçiminde oluşturulan yoğunluk haritasından kaynaklanmaktadır. Fakat ön işlem adımı çevrimdışı yapıldığı için bunun sorgu süresine bir etkisi yoktur ve önerilen yöntem veri kümesi önceden verilen ve yüksek doğruluk ve hızlı sorgu zamanı gerektiren uygulamalar için uygundur.

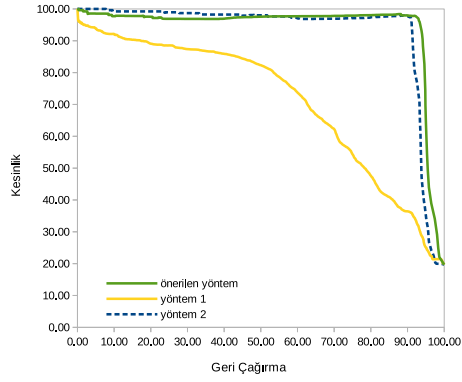
Tablo 4.1'e ek olarak Şekil 4.4'de yapılan tüm testler için keskinlik-geri çağırma grafikleri verilmiştir. Bu grafiklerden de görüleceği üzere %90 üzeri geri çağırma- larda önerilen yöntem yöntem 2'den daha iyi sonuçlar vermiştir.



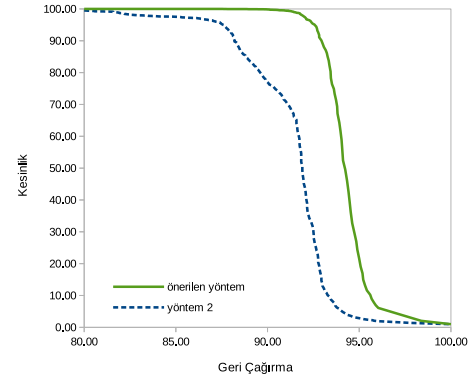
Çizelge 4.1: Deneysel sonuçlar

Test #	GS <sup>1</sup>	SS <sup>1</sup>	ÖS (K) <sup>1</sup>	Yöntem	GÇ <sup>1</sup> [%]	Kesinlik [%]	F-değeri	SZ <sup>1</sup> [s/sorgu]	ÖZ <sup>1</sup> [s]
1	255	255	hepsi	yöntem 1	62.24	71.17	66.41	1780.68	65.11
1	255	255	50	yöntem 2	90.84	97.31	93.96	2.29	25.56
1	255	255	50	önerilen yöntem	92.84	97.13	94.94	1.91	135.13
2	5100	100	10	yöntem 2	87.01	96.35	91.44	2.94	700.03
2	5100	100	10	önerilen yöntem	91.30	99.28	95.12	2.67	4065.96
3	5100	100	50	yöntem 2	90.44	99.22	94.63	58.41	773.08
3	5100	100	50	önerilen yöntem	95.28	99.73	97.45	51.67	4177.64
4	5100	100	100	yöntem 2	91.92	96.26	94.04	200.66	586.27
4	5100	100	100	önerilen yöntem	94.42	99.61	96.95	168.52	4085.45
5	5100	5100	50	yöntem 2	84.80	92.72	88.58	58.57	765.84
5	5100	5100	50	önerilen yöntem	89.73	95.83	92.68	52.58	4324.73
6	25500	500	50	yöntem 2	90.40	99.22	94.60	223.32	3890.35
6	25500	500	50	önerilen yöntem	94.89	99.66	97.22	205.65	18688.37

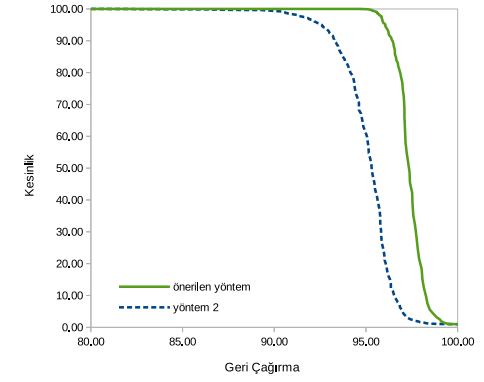
<sup>1</sup> GS: Görüntü Sayısı, SS: Sorgu Sayısı, ÖS: Öznitelik Sayısı, GÇ: Geri Çağırma, SZ: Sorgu Zamanı, ÖZ: Ön İşlem Zamanı



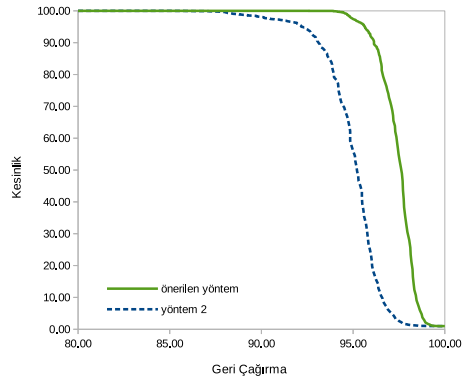
(a)



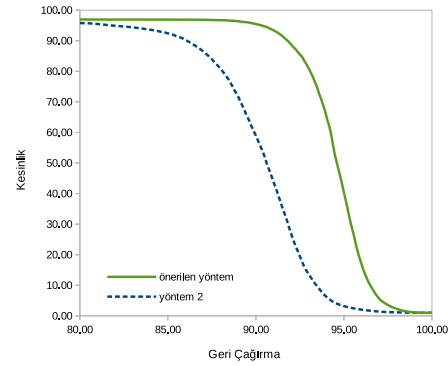
(b)



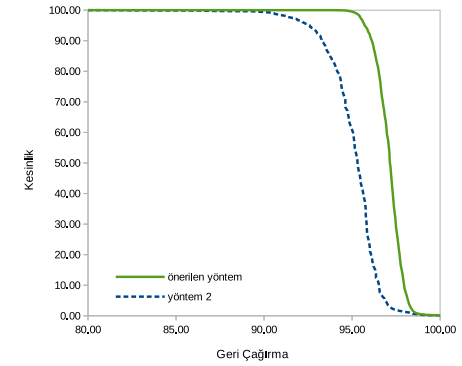
(c)



(d)



(e)



(f)

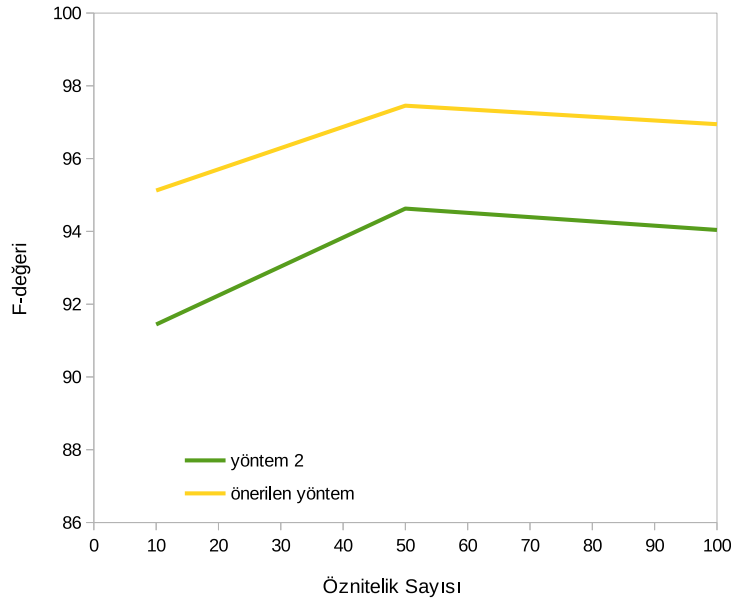
Şekil 4.4: Yapılan her bir test için kesinlik-geri çağırma grafiği (a) Test 1 için kesinlik-geri çağırma grafiği (b) Test 2 için kesinlik-geri çağırma grafiği (c) Test 3 için kesinlik-geri çağırma grafiği (d) Test 4 için kesinlik-geri çağırma grafiği (e) Test 5 için kesinlik-Geri çağırma grafiği (f) Test 6 için kesinlik-geri çağırma grafiği



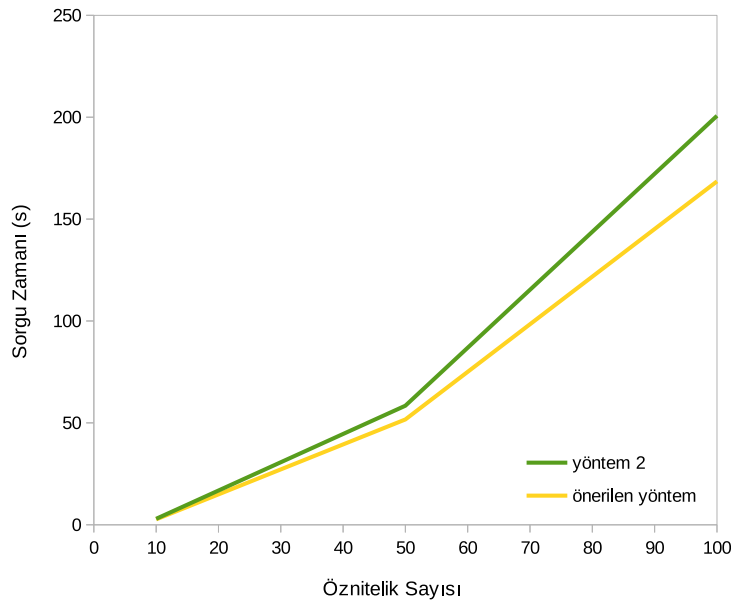
## 4.6 Seçilen Öznitelik Sayısının Sonuçlara Etkisi

Neredeyse özdeş tespitinde oluşturulan öznitelik alt kümesinin sonuçlar üzerinde büyük etkisi vardır. Şekil 4.5a'de F-değeri-öznitelik sayısı grafiği verilmiştir ve bu grafikten görüleceği üzere öznitelik sayısının optimum bir değeri vardır ve burada bu değer 50'dir. İlk bakışta öznitelik sayısının artmasıyla F-değerinin sürekli artacağı düşünülmektedir. Fakat öznitelik kümeleri arasındaki uzaklığı hesaplamada kullanılan EMD algoritmasından dolayı öznitelik sayısının belirli bir değerin üzerinde olması uzaklık hesabında fazla gürültü oluşturduğundan dolayı öznitelik sayısının optimum bir değeri oluşmuştur.

Seçilen öznitelik sayısının sorgu zamanına etkisini görmek için ise sorgu zamanı-öznitelik sayısı grafiği çizilmiş ve bu grafik Şekil 4.5b'de verilmiştir. Burada sorgu süresindeki artışın EMD algoritmasının çalışma zamanına bağlı olduğunu hatırlamakta fayda vardır ve kullanılan FastEMD algoritmasının çalışma zamanı eşiklenen değer sayısı ile doğrudan değiştiği için öznitelik sayısındaki artış ile sorgu zamanı arasında sağlıklı bir orantı kurulamamıştır.



(a)



(b)

Şekil 4.5: Öznitelik sayısına göre F-değeri ve sorgu zamanı değişimleri (a) F-değerine karşılık öznitelik sayısı grafiği (b) Sorgu zamanına karşılık öznitelik sayısı grafiği

#### 4.7 Dönüşüm Türlerinin Sonuçlara Etkisi

Bu alt bölümde önerilen yöntem ve yöntem 2 dönüşüm türlerinin sonuçlara etkisi yönüyle karşılaştırılacaktır. Bunun için öncelikle orijinal görüntüler üzerinde neredeyse özdeşler elde edilmek için yapılan ve Sayfa 22'de listesi verilen dönüşümler 15 grup altında toplanmıştır. Bu gruplar ve bu gruplara dahil olan dönüşümler aşağıdaki listede verilmiştir.

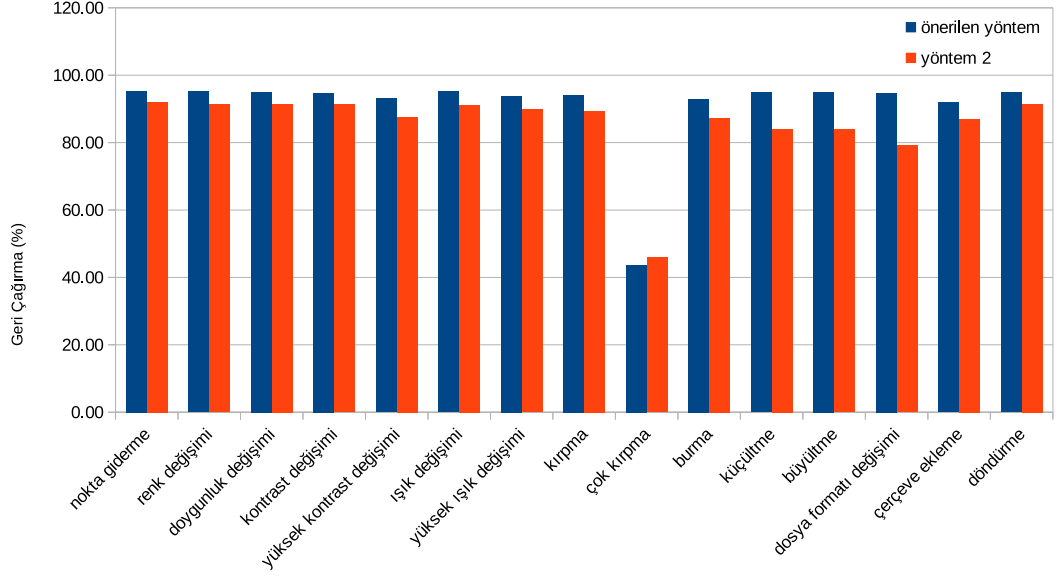
- **Nokta giderme:** Bu gruba sadece siyah nokta giderilerek elde edilen neredeyse özdeş dahil edilmiştir.
- **Renk değişimi:** Bu gruba kırmızı, yeşil ve mavi renk üzerinde %10'luk bir değişim yapılarak elde edilen 3 neredeyse özdeşler dahil edilmiştir.
- **Doğunluk değişimi:** Bu gruba doğunluk değişimi ile elde edilen neredeyse özdeşler dahil edilmiştir.
- **Kontrast değişimi:** Bu gruba kontrast değişimi ile elde edilen neredeyse özdeşler dahil edilmiştir. Buradaki kontrast değişimi görüntü üzerindeki her bir noktanın kontrastını %10 artırma ve yine %10 azaltmadır.
- **Yüksek kontrast değişimi:** Bu gruba yüksek kontrast değişimi ile elde edilen neredeyse özdeşler dahil edilmiştir. Bu kontrast değişimleri orijinal görüntünün her bir noktasının kontrast değerlerini 3 katına çıkarma ve 1/3 katına indirmedir.
- **Işık değişimi:** Bu gruba ışık değişimi ile elde edilen neredeyse özdeşler dahil edilmiştir. Buradaki ışık değişimleri ışık seviyesini %10 ve %20 azaltma ve artırmadır.
- **Yüksek ışık değişimi:** Bu gruba yüksek ışık değişimi ile elde edilen neredeyse özdeşler dahil edilmiştir. Buradaki ışık değişimleri %50 azaltma ve %50 artırmadır.
- **Kırpma:** Bu gruba görüntü üzerinde küçük oranlarda yapılan kırpma ile elde edilen neredeyse özdeşler dahil edilmiştir. Buradaki oranlar %5, %10, %20 ve %30'dur. Bu kırpma görüntüye en dıştan başlanarak uygulanmıştır.
- **Çok kırpma:** Bu gruba görüntü üzerinde büyük oranlarda yapılan kırpma ile elde edilen neredeyse özdeşler dahil edilmiştir. Buradaki oranlar %50, %70 ve %90'dır. Bu kırpma da aynı şekilde görüntüye en dıştan başlanarak uygulanmıştır.

- **Burma:** Bu gruba görüntüye sırasıyla %5, %10 ve %15 burma uygulanarak elde edilen neredeyse özdeşler dahil edilmiştir.
- **Küçültme:** Bu gruba hem örnekleme ile küçültülerek elde edilen neredeyse özdeşler hem de küçültme işleminden sonra tekrar eski boyutuna getirilerek elde edilen neredeyse özdeşler dahil edilmiştir.
- **Büyültme:** Bu gruba büyültme işleminden sonra tekrar orijinal boyutuna getirilerek elde edilen neredeyse özdeşler dahil edilmiştir.
- **Dosya formatı değişimi:** Bu gruba sadece jpeg'den gif formatına dönüştürülerek elde edilen neredeyse özdeş dahil edilmiştir.
- **Çerçeve ekleme:** Bu gruba orijinal görüntüye dış çerçeve eklenerek elde edilen neredeyse özdeşler dahil edilmiştir.
- **Döndürme:** Bu gruba ise orijinal görüntüyü döndürerek elde edilen neredeyse özdeşler dahil edilmiştir.

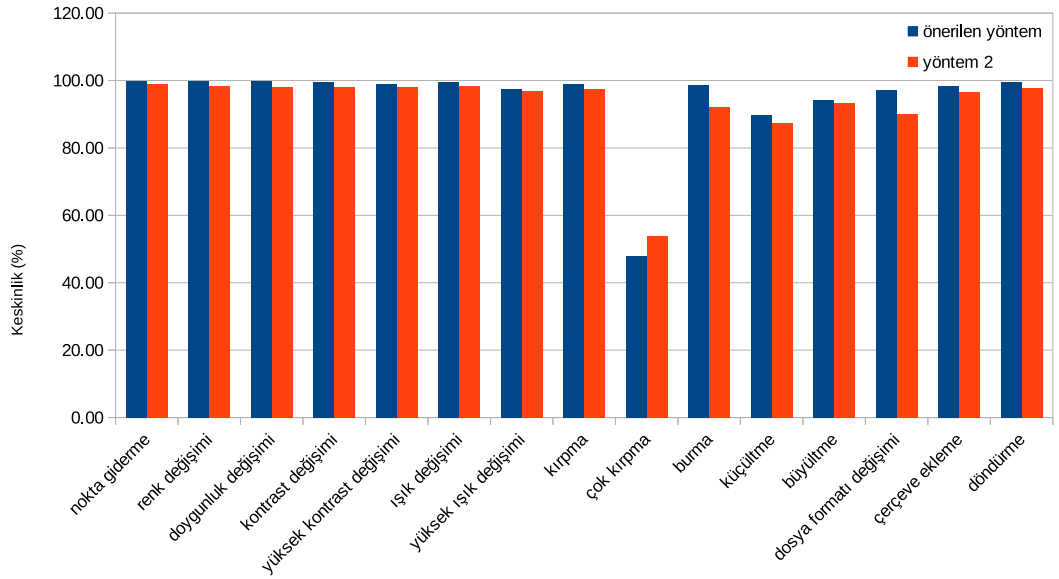
Şekil 4.6'de üstte verilen dönüşüm türü gruplarına göre ortalama geri çağırma grafiği vermiştir. Bu geri çağırma değerleri 5. test sonuçlarından en yüksek F-değeri'ne denk gelen sonuçlardır. Grafikten de görüleceği üzere karşılaştırılan her iki yöntem içinde en kötü geri çağırma sonucu yüksek oranlarda yapılan kırpma işlemi ile elde edilen neredeyse özdeşlerden gelen sonuçlardır. Bu nokta aynı zamanda yöntem 2'nin önerilen yöntemden daha iyi olduğu tek noktadır. Yüksek oranlarda yapılan kırpma işlemleri orijinal görüntünün büyük bir bölümünün atılmasına ve dolayısıyla elde edilen yeni görüntü ile orijinal görüntünün çok az ortak bölgesi kalmasına sebep olur. Buda bu görüntülerin çok daha az sayıda ortak öz-niteliğe sahip olması manasına gelir. Bunun da sonuçları kötü etkilemesi beklenen bir durumdur.

Şekil 4.7'de ise dönüşüm türü gruplarına göre ortalama keskinlik grafiği verilmiştir. Bu grafiğe göre önerilen yöntem her bir grup için yöntem 2'den daha iyi sonuç vermiştir. Yine burada da büyük kırpma oranlarında her iki yöntemde kötü sonuç vermiştir.

Bu sonuçlardan gözlemlenen bir diğer sonuç ise fotometrik dönüşümlerde önerilen yöntemin yöntem 2'den daha iyi sonuçlar vermesidir. Bunun sebebi önerilen yöntemde kullanılan yoğunluk haritasının fotometrik değişimlere karşı daha dirençli olmasıdır. Diğer yandan yöntem 2'de öz-nitelikler sadece öz-niteliğin görüntü üzerinde bulunduğu noktanın ışık değerine göre seçilmiştir ve ışık değeri fotometrik dönüşümler ile kolayca değişmektedir.



Şekil 4.6: Dönüşüm türlerine göre en iyi F-değeri için ortalama geri çağırma değerleri



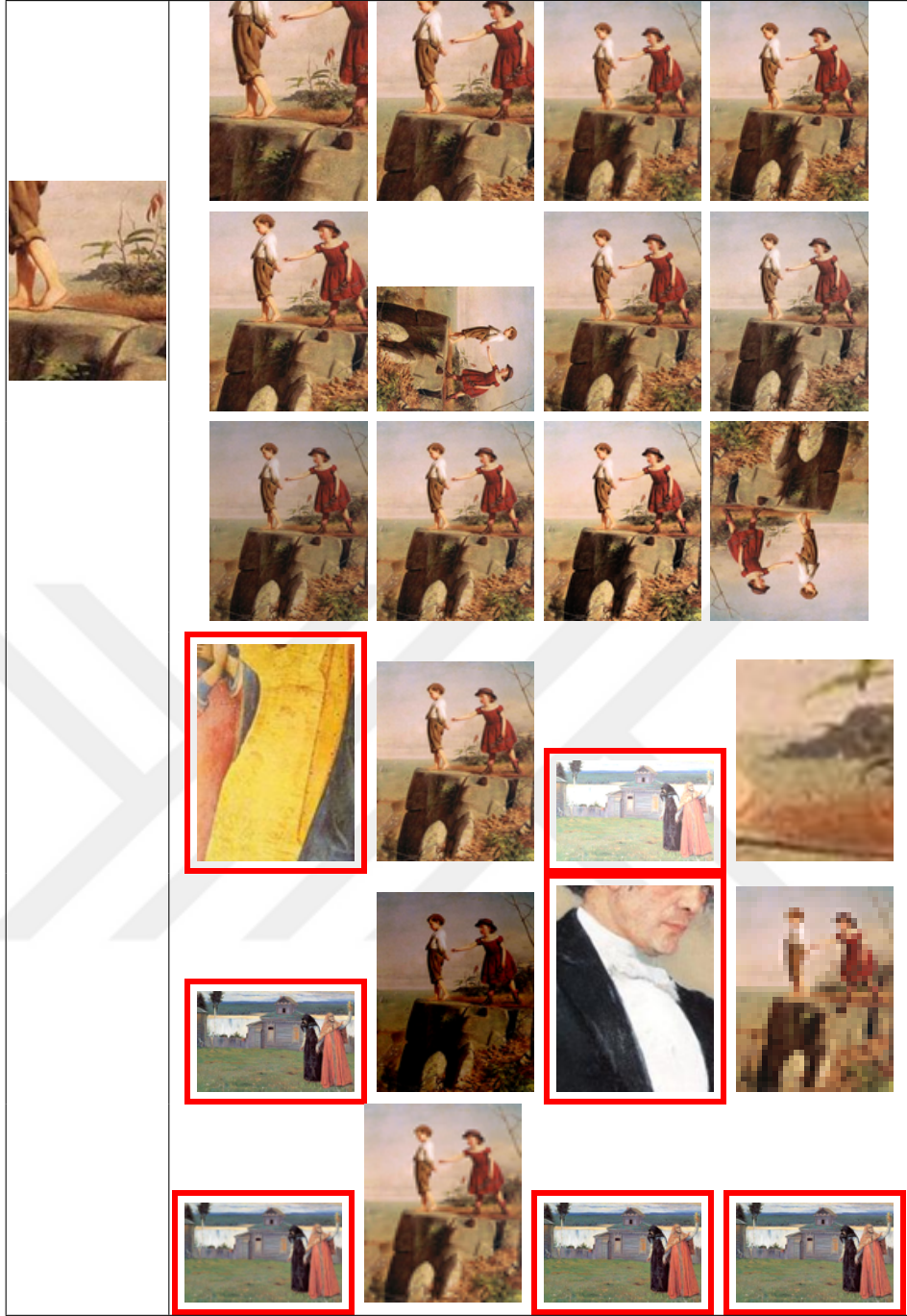
Şekil 4.7: Dönüşüm türlerine göre en iyi F-değeri için ortalama keskinlik değerleri

#### 4.8 Sonuçları Etkileyen Diğer Etkenler

Görüntüden çıkarılan tüm özniteliklerin sayısı keskinlik ve geri çağırma metriklerini etkilemektedir. Özellikle öznitelik sayısı alt küme sayısında küçük ise bu tür görüntüler neredeyse özdeşi olmayan görüntülerle eşleşebilmektedir. Bununla birlikte özellikle büyük oranda kırılmış görüntülerde, hem öznitelik sayısı oldukça düşmekte hem de yoğunluk haritası neredeyse özdeşlerle örtüşmeyebilmektedir. Dolayısıyla görüntünün kırılması da sonuçları olumsuz etkilemektedir. Bu tür gö-

rüntülere bir örnek ve bu örnek için 5. test sonuçlarından elde edilen ve neredeyse özdeş olarak işaretlenen ilk 24 görüntü Şekil 4.4'de verilmiştir. Şekilde verilen sonuç, önerilen yöntemin sonucudur. Sorguda kullanılan görüntüden toplamda elde edilen öznitelik sayısı 624'tür ve bu sayı öznitelik sayısı ortalamasının oldukça altındadır. Aynı zamanda bu görüntü orijinal görüntünün %70'i kırılmış halidir. Sorgu sonucunun ilk 24 görüntüsüne bakıldığında 7 tane yanlış pozitif olduğu görülmektedir.





Resim 4.4: Öznitelik sayısı az olan ve orijinalinin %70'i kırılmış bir görüntü için sorgu görüntüsüyle birlikte sorgu sonucundan sıralı olarak ilk 24 görüntü. Yanlış pozitif olanlar kırmızı çerçeve içine alınmıştır.





## 5. SONUÇ VE ÖNERİLER

Bu tezde neredeyse özdeş tespiti için yeni bir öznitelik seçme yöntemi sunulmuştur. Bu yöntemde öznitelikler kullanılarak bir yoğunluk haritası oluşturulmuş ve öznitelik seçme işlemi bu oluşturulan yoğunluk haritası ile yapılmıştır. Oluşturulan yoğunluk haritası ile özniteliklerin birbirini destekleyerek tekrar olasılığı yüksek özniteliklerin belirlenmesi amaçlanmıştır. Yapılan testler önerilen yöntemin ön işlem süresindeki artışla birlikte önceki çalışmalardan daha iyi sonuçlar verdiğini göstermektedir. Daha özelde tüm öznitelikler kullanılarak yapılan tespitin önerilen yöntemden daha iyi sonuçlar vermesi beklenirken, görüntülerden elde edilen özniteliklerin sayısının çok fazla olması eşleştirmede kullanılan EMD algoritması için çok fazla gürültü oluşturmaktadır ve bu nedenle önerilen yöntem beklenenin aksine çok daha iyi sonuçlar vermektedir. Önerilen yöntemin sadece ışık seviyesine göre öznitelik seçimi yaparak neredeyse özdeş tespiti yapan yöntemden daha iyi sonuç vermesinin sebebi ise görüntüde yapılan değişikliklerin görüntü üzerindeki ışık seviyelerini değiştirmesi ve buna bağlı olarak seçilen özniteliklerin neredeyse özdeşler arasında farklılık göstermesidir.

Yapılan testlerde EMD ile uzaklık hesabında özniteliklerin görüntü üzerindeki konumları kullanılmadı ve bu sebepten dolayı bir sorgu görüntüsü için elde sonuçlar geometrik bir doğrulamadan geçmemiş oldu. Dolayısıyla halihazırda elde edilen sonuçlar farklı geometrik eşleştirme yöntemleri (Örn. RANSAC) kullanılarak sonuçlar daha da iyileştirilebilir.

Yine yapılan testlerde görüntülerden elde edilen SIFT öznitelikleri ve orijinal SIFT tanımlayıcıları kullanılmıştır. Orijinal SIFT tanımlayıcıları 128 boyutludur ve EMD uzaklığı hesaplanırken her bir öznitelik arasındaki uzaklık bu 128 boyutluk tanımlayıcılar ile hesaplanır. Bu ise sorgu zamanının uzamasına sebep olmaktadır. Ke ve arkadaşlarının yaptığı çalışmanın [21] sonuçlarına göre Temel Bileşen Analizi yapılarak 36 boyuta düşürülen SIFT tanımlayıcısı daha az boyutla orijinal tanımlayıcıya göre daha iyi sonuçlar vermiştir. Dolayısıyla elde edilen F-değerleri ve sorgu zamanları bu tanımlayıcı ile biraz daha iyileştirilebilir. Bunun yanında SUFT ve FAST gibi SIFT'ten farklı yerel özniteliklerle de elde edilen sonuçlar desteklenebilir.

Özellikle büyük bir bölümü kırılmış ve az sayıda öznelik bulunan görüntülerde önerilen yöntemin çok etkili olamadığı görülmüştür. Bu tip görüntülerde önerilen yönteme yardımcı olarak makine öğrenmesi yöntemlerinden faydalanılarak tek başına yoğunluk haritası ile oluşturulan öznelik alt kümesinden daha iyi sonuçlar verebilecek alt kümeler oluşturulabilir. Bu şekilde yöntemin zayıf olduğu bu durum giderilmiş olur.

Önerilen yöntem aynı zamanda MapReduce [14] gibi dağıtık sistemlere de çok kolay bir şekilde entegre edilebilir. Örneğin seçilen öznelikler dağıtık yapıda dosya saklayabilen HDFS [39] dosya sistemine aktarılabilir. Daha sonra bir MapReduce işi yazılarak, işin map kısmında sorgu görüntüsünün veri kümesindeki görüntüler ile EMD uzaklıkları hesaplanır. Reduce kısmında ise hesaplanan bu uzaklıklar eşiklenip görüntünün neredeyse özdeşleri bulunabilir. Bu şekilde çok büyük veri kümeleri üzerinde sorgular çalıştırılabilir.

Önerilen yöntemin kısıtlarından biri EMD algoritmasının veri kümesindeki tüm görüntüler için tekrar tekrar çalıştırılması ve bundan kaynaklı sorgu zamanının fazla olmasıdır. Bu kısıt veri kümesindeki görüntülere LSH [22] vb. bir ön filtreleme uygulanarak aşılabilir veya EMD yerine Bölüm 2.4'da bahsedilen diğer uzaklık hesaplama yöntemlerinden birisi kullanılabilir. Fakat EMD yerine başka bir uzaklık yöntemi kullanmak yöntemi EMD'nin hataya karşı dirençli olması özelliğinden de mahrum bırakacaktır. Burada bir ödün verme durumu söz konusudur ve bu durum eniyilenebilir.

Yöntemin kısıtlarından bir diğeri ise sorgulama işleminden önce veri kümesindeki her bir görüntüden seçilen özneliklerin belleğe alınması gerekliliğidir. Bu ise, çok büyük veri kümelerinde yöntemin bu haliyle uygulanamaması manasına gelmektedir. Bu kısıt seçilen özneliklerin sorgulama esnasında diskten okunmasıyla aşılabilir fakat bu defa da diskten okumanın yavaşlığı sorgu süresini olumsuz etkileyecektir.

Önerilen yöntemin üstte bahsedilen kısıtlarına karşın, karşılaştırılan diğer yöntemler de aynı kısıtlara sahiptirler ve şu anki durumda önerilen yöntem deneysel sonuçlardan da görüldüğü üzere daha iyi sonuçlar vermektedir.

## KAYNAKLAR

- [1] **Foo, J. J.** ve **Sinha, R.** “Pruning SIFT for scalable near-duplicate image matching”. In: *Proceedings of the eighteenth conference on Australasian database-Volume 63*. Australian Computer Society, Inc. 2007, pp. 63–71.
- [2] **O’Ruanaidh, J.** ve **Pun, T.** “Rotation, translation and scale invariant digital image watermarking”. In: *Image Processing, International Conference on*. Vol. 1. IEEE Computer Society. 1997, pp. 536–536.
- [3] **Kundur, D.** ve **Hatzinakos, D.** “Digital watermarking for telltale tamper proofing and authentication”. In: *Proceedings of the IEEE 87.7* (1999), pp. 1167–1180.
- [4] **Podilchuk, C. I.** ve **Delp, E. J.** “Digital watermarking: algorithms and applications”. In: *Signal Processing Magazine, IEEE* 18.4 (2001), pp. 33–46.
- [5] **Lin, P. L., Hsieh, C.-K.,** ve **Huang, P.-W.** “A hierarchical digital watermarking method for image tamper detection and recovery”. In: *Pattern recognition* 38.12 (2005), pp. 2519–2529.
- [6] **Chen, L.** ve **Stentiford, F.** “Comparison of near-duplicate image matching”. In: *Proceedings of the 3rd European Conference on Visual Media Production*. 2006.
- [7] **Chang, E. Y.** ve diğerleri. “Rime: A replicated image detector for the world wide web”. In: *Photonics East (ISAM, VVDC, IEMB)*. International Society for Optics ve Photonics. 1998, pp. 58–67.
- [8] **Smeulders, A. W.** ve diğerleri. “Content-based image retrieval at the end of the early years”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.12 (2000), pp. 1349–1380.
- [9] **Fridrich, A. J., Soukal, B. D.,** ve **Lukáš, A. J.** “Detection of copy-move forgery in digital images”. In: *in Proceedings of Digital Forensic Research Workshop*. Citeseer. 2003.
- [10] **Luo, J.** ve **Nascimento, M. A.** “Content based sub-image retrieval via hierarchical tree matching”. In: *Proceedings of the 1st ACM international workshop on Multimedia databases*. ACM. 2003, pp. 63–69.

- [11] **Meng, Y., Chang, E., ve Li, B.** “Enhancing dpf for near-replica image recognition”. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2003, pp. II–416.
- [12] **Zheng, L.** ve diğerleri. “Near-duplicate image detection in a visually salient riemannian space”. In: *Information Forensics and Security, IEEE Transactions on* 7.5 (2012), pp. 1578–1593.
- [13] **Kim, S.** ve diğerleri. “Near Duplicate Image Discovery on One Billion Images”. In: *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE. 2015, pp. 943–950.
- [14] **Dean, J.** ve **Ghemawat, S.** “MapReduce: simplified data processing on large clusters”. In: *Communications of the ACM* 51.1 (2008), pp. 107–113.
- [15] **Ke, Y.** ve diğerleri. “Efficient near-duplicate detection and sub-image retrieval”. In: *ACM Multimedia*. Vol. 4. 1. 2004, p. 5.
- [16] **Jinde, W.** ve diğerleri. “A SIFT Pruning Algorithm for Efficient Near-Duplicate Image Matching [J]”. In: *Journal of computer-aided design & computer graphics* 6 (2010), p. 021.
- [17] **Zhao, W.-L.** ve **Ngo, C.-W.** “Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection”. In: *Image Processing, IEEE Transactions on* 18.2 (2009), pp. 412–423.
- [18] **Lowe, D. G.** “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.
- [19] **Bay, H., Tuytelaars, T., ve Van Gool, L.** “Surf: Speeded up robust features”. In: *Computer vision–ECCV 2006*. Springer, 2006, pp. 404–417.
- [20] **Rosten, E.** ve **Drummond, T.** “Machine learning for high-speed corner detection”. In: *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.
- [21] **Ke, Y.** ve **Sukthankar, R.** “PCA-SIFT: A more distinctive representation for local image descriptors”. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2004, pp. II–506.
- [22] **Indyk, P.** ve **Motwani, R.** “Approximate nearest neighbors: towards removing the curse of dimensionality”. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM. 1998, pp. 604–613.
- [23] **Fischler, M. A.** ve **Bolles, R. C.** “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

- [24] **Topsøe, F.** “Some inequalities for information divergence and related measures of discrimination”. In: *Information Theory, IEEE Transactions on* 46.4 (2000), pp. 1602–1609.
- [25] **Huttenlocher, D. P., Klanderman, G. A., ve Rucklidge, W. J.** “Comparing images using the Hausdorff distance”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 15.9 (1993), pp. 850–863.
- [26] **Dubuisson, M.-P. ve Jain, A. K.** “A modified Hausdorff distance for object matching”. In: *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*. Vol. 1. IEEE. 1994, pp. 566–568.
- [27] **Rucklidge, W.** *Efficient visual recognition using the Hausdorff distance*. Vol. 1173. Springer Heidelberg, 1996.
- [28] **Jesorsky, O., Kirchberg, K. J., ve Frischholz, R. W.** “Robust face detection using the hausdorff distance”. In: *Audio-and video-based biometric person authentication*. Springer. 2001, pp. 90–95.
- [29] **Aspert, N., Santa Cruz, D., ve Ebrahimi, T.** “MESH: measuring errors between surfaces using the Hausdorff distance.” In: *ICME (1)*. 2002, pp. 705–708.
- [30] **Vaserstein, L. N.** “Markov processes over denumerable products of spaces, describing large systems of automata”. In: *Problemy Peredachi Informat-sii* 5.3 (1969), pp. 64–72.
- [31] **Rubner, Y., Tomasi, C., ve Guibas, L. J.** “A metric for distributions with applications to image databases”. In: *Computer Vision, 1998. Sixth International Conference on*. IEEE. 1998, pp. 59–66.
- [32] **Rubner, Y., Tomasi, C., ve Guibas, L. J.** “The earth mover’s distance as a metric for image retrieval”. In: *International journal of computer vision* 40.2 (2000), pp. 99–121.
- [33] **Pele, O. ve Werman, M.** “Fast and robust earth mover’s distances”. In: *Computer vision, 2009 IEEE 12th international conference on*. IEEE. 2009, pp. 460–467.
- [34] **Walker, K. N., Cootes, T. F., ve Taylor, C. J.** “Locating salient facial features using image invariants”. In: *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE. 1998, pp. 242–247.
- [35] **Url-1.** *CGFA Art Gallery*. 1996. URL: <http://ftp.icm.edu.pl/cgfa/> (alındığı tarih 11/30/2015).
- [36] **ImageMagick Studio, L.** *ImageMagick*. 2008.

- [37] **Tange, O.** ve diğeri. “Gnu parallel-the command-line power tool”. In: *The USENIX Magazine* 36.1 (2011), pp. 42–47.
- [38] **Bradski, G.** ve diğeri. “The opencv library”. In: *Doctor Dobbs Journal* 25.11 (2000), pp. 120–126.
- [39] **Borthakur, D.** “HDFS architecture guide”. In: *HADOOP APACHE PROJECT* [http://hadoop.apache.org/common/docs/current/hdfs\\_design.pdf](http://hadoop.apache.org/common/docs/current/hdfs_design.pdf) (2008).



## ÖZGEÇMİŞ

**Ad-Soyad** : Burak YILDIZ  
**Uyruđu** : TC  
**Dođum Tarihi ve Yeri** : 29 Ekim 1988 Kırřehir  
**E-posta** : burakyildiz@etu.edu.tr

### ÖĐRENİM DURUMU:

- **Lisans** : 2013, TOBB ETÜ, Mühendislik Fakültesi, Makine Mühendisliđi

### MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2008-2009	Rona Makine Sanayi ve Ticaret A.ř.	Stajyer
2011-2012	Toyota Otomotiv Sanayi Türkiye A.ř	Stajyer
2012-2013	TOBB Ekonomi ve Teknoloji Üniversitesi	Stajyer
2013-2014	TOBB Ekonomi ve Teknoloji Üniversitesi	Tam Burslu Asistan
2014-...	Turgut Özal Üniversitesi	Arařtırma Görevlisi

**YABANCI DİL:** İngilizce, Almanca

### TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Yıldız, B.**, ve Demirci, M. F., “Distinctive interest point selection for efficient near-duplicate detection,” in SSIAl. IEEE, 2016, pp. 49-52.
- **Yıldız, B.**, ve Demirci, M. F., “Efficient near-duplicate image detection with created feature subset,” in SIU. IEEE, 2016, pp. –.