

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**SAYISAL GÖRÜNTÜLERDE PİKSEL YOLU  
ÇIKARMA ESASLI BOYUT DEĞİŞİKLİĞİ TESPİTİ**

**DOKTORA TEZİ**

**Zehra KARAPINAR ŞENTÜRK**

**Enstitü Anabilim Dalı** : **BİLGİSAYAR VE BİLİŞİM  
MÜHENDİSLİĞİ**

**Tez Danışmanı** : **Yrd. Doç. Dr. Devrim AKGÜN**

**Aralık 2016**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

SAYISAL GÖRÜNTÜLERDE PİKSEL YOLU  
ÇIKARMA ESASLI BOYUT DEĞİŞİKLİĞİ TESPİTİ


DOKTORA TEZİ


Zehra KARAPINAR ŞENTÜRK

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM  
MÜHENDİSLİĞİ


Bu tez 01 / 12 / 2016 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

  
Prof. Dr.  
Recep DEMİRCİ  
Jüri Başkanı

  
Doç. Dr.  
Uğur GÜVENÇ  
Üye

  
Doç. Dr.  
Pakize ERDOĞMUŞ  
Üye

  
Doç. Dr.  
Cüneyt BAYILMIŞ  
Üye

  
Yrd. Doç. Dr.  
Devrim AKGÜN  
Üye

## **BEYAN**

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Zehra KARAPINAR ŞENTÜRK

01.12.2016

## TEŐEKKÜR

Doktora eđitimim boyunca deđerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteđini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren deđerli danışman hocam Yrd. Doç. Dr. Devrim AKGÜN'e teşekkürlerimi sunarım.

Eđitimim boyunca benden maddi ve manevi desteklerini esirgemeyen, çalışmam için hep uygun koşulları sağlamaya çalışan, beni hiç yalnız bırakmayan sevgili eşime, bana hep güvenen aileme, yaşadığım tüm zorluklara rağmen bana güç veren, hayatımı anlamlandıran, mutluluk kaynađım, biricik kızım Beyza'ma da çok teşekkür ederim.

Ayrıca bu çalışmanın maddi açıdan desteklenmesine olanak sağlayan Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığına (Proje No: 2015-50-02-019) teşekkür ederim.

## İÇİNDEKİLER

TEŞEKKÜR .....	i
İÇİNDEKİLER .....	ii
SİMGELER VE KISALTMALAR LİSTESİ .....	iv
ŞEKİLLER LİSTESİ .....	v
TABLolar LİSTESİ .....	vii
ÖZET .....	viii
SUMMARY .....	ix
BÖLÜM 1.	
GİRİŞ .....	1
1.1. Görsel Kalitenin İyileştirilmesi .....	3
1.2. İşlem Hızının Artırılması .....	6
1.3. Piksel Yolu Çıkarma Tabanlı Ölçeklemenin Tespiti .....	7
BÖLÜM 2.	
PİKSEL YOLU ÇIKARMA (SEAM CARVING) VE SINIFLAYICILAR .....	13
2.1. Piksel Yolu Çıkarma (Seam Carving) .....	13
2.2. Yerel İkili Örüntüler (YİÖ) .....	25
2.3. Destek Vektör Makineleri (DVM) .....	31
2.3.1. Lineer bir sınıflayıcı ve lineer ayrılabilen problem .....	33
2.3.2. Lineer bir sınıflayıcı ve tamamen lineer ayrılamayan problem .....	37
2.3.3. Lineer olmayan sınıflayıcı ve lineer ayrılamayan problem .....	39
2.4. Izgara Arama (Grid Search) Yöntemi .....	40
2.4.1. GPU ile hızlandırılmış ızgara arama .....	42
2.5. K Katlı Çarpaz Sağlama (K-Fold Cross Validation) .....	45

2.6. ROC (Receiver Operating Characteristics) Analizi.....	46
BÖLÜM 3.	
PİKSEL YOLU ÇIKARMA İÇİN HİBRİT VE ŞERİT TABANLI TESPİT	50
3.1. Hibrit Pıksel Yolu Çıkarma Tespit Yöntemi.....	52
3.1.1. Çıkarılan özellikler .....	53
3.2. Şerit Tabanlı Pıksel Yolu Çıkarma Tespit Yöntemi .....	58
BÖLÜM 4.	
DENEYSEL SONUÇLAR VE TARTIŞMA .....	64
4.1. Deneysel Altyapı .....	64
4.2. Hibrit Yöntem ile Tespit Sonuçları .....	66
4.3. Şerit Tabanlı Yöntem ile Tespit Sonuçları .....	67
4.4. Tartışma .....	76
BÖLÜM 5.	
SONUÇ VE ÖNERİLER .....	79
KAYNAKLAR .....	81
ÖZGEÇMİŞ .....	89

## SİMGELER VE KISALTMALAR LİSTESİ

AKD	: Ayrık Kosinüs Dönüşümü
ADF	: Ağırlıklı Dağıtım Fonksiyonu
CTM	: Close To Metal
CUDA	: Compute Unified Device Architecture
DN	: Doğru Negatif sayısı
DNO	: Doğru Negatif Oranı
DP	: Doğru Pozitif sayısı
DPO	: Doğru Pozitif Oranı
DVM	: Destek Vektör Makinesi
GB	: Giga Byte
ICC	: Interveining Contour and Color
Kds	: Dikey Seam'in KMEM'si
KMEM	: Kümülatif Minimum Enerji Matrisi
Kys	: Yatay Seam'in KMEM'si
maks	: Maksimum
min	: Minimum
QF	: Quality Factor
ROC	: Receiver Operating Characteristics
SIFT	: Scale Invariant Feature Transform
std	: Standart Sapma
SURF	: Speed-Up Robust Feature
UCID	: Uncompressed Image Dataset
YİÖ	: Yerel İkili Örüntüler
YN	: Yanlış Negatif sayısı

YNO : Yanlıř Negatif Oranı  
YP : Yanlıř Pozitif sayısı  
YPO : Yanlıř Pozitif Oranı



## ŞEKİLLER LİSTESİ

Şekil 2.1. Orijinal görüntü ve enerji haritası üzerinde yatay ve dikey piksel yolları.....	13
Şekil 2.2. Piksel yolu çıkarma algoritması (genişlik azaltma).....	15
Şekil 2.3. Optimal piksel yolunun bulunması .....	17
Şekil 2.4. Görüntünün genişliğinin azaltılmasına yönelik beş farklı yaklaşım: klasik ölçekleme, kırpma, sütun çıkarma, piksel çıkarma ve piksel yolu çıkarma .....	18
Şekil 2.5. Orijinal görüntü, piksel yolu çıkarma yöntemi ile %30 ölçeklenmiş görüntü ve klasik (eş oranlı) ölçekleme ile %30 ölçeklenmiş görüntü ..	21
Şekil 2.6. Örnek piksel için YİÖ dönüşümü .....	26
Şekil 2.7. Farklı komşuluk mesafesine ve komşu sayısına sahip YİÖ örnekleri ....	27
Şekil 2.8. Piksel yolu çıkarma öncesi ve sonrası piksel yoluna komşu piksellerin YİÖ değerlerindeki değişim .....	27
Şekil 2.9. Gri görüntü (solda) ve YİÖ dönüşümü (sağda) .....	28
Şekil 2.10. (a) İki sınıflı veri kümesi ve muhtemel ayırıcı düzlemler (b) DVM'nin "zor noktalar"a (kırmızı çerçevesi) göre belirlediği optimal hiper düzlem .....	32
Şekil 2.11. Lineer ayrılabilen iki sınıfı ayıran hiper düzlem.....	34
Şekil 2.12. Tamamen lineer ayrılamayan veri kümesi ve lineer ayıran hiper düzlem .....	37
Şekil 2.13. Radyal Temelli Çekirdek ile yeniden haritalanarak ikiye ayrılmış veri	39
Şekil 2.14. (c,y) değiştiğinde doğruluktaki değişim .....	41
Şekil 2.15. CUDA mimarisi .....	43
Şekil 2.16. CPU-GPU .....	43
Şekil 2.17. CUDA program akışı .....	44
Şekil 2.18. Farklı Cost ve Gamma değerleri için GPU ile elde edilen hızlandırma	45

Şekil 2.19. Dört Katlı Çapraz Sağlama .....	46
Şekil 2.20. Karmaşıklık matrisi.....	48
Şekil 2.21. Beş ayrık sınıflandırıcıyı gösteren ROC eğrisi .....	49
Şekil 2.22. Örnek ROC eğrisi .....	49
Şekil 3.1. Piksel yolu çıkarma yöntemi ile ölçeklenmiş görüntüler.....	51
Şekil 3.2. Senaryo 1: 18 özellik ile eğitim ve test.....	55
Şekil 3.3. Senaryo 2: 24 özellik ile eğitim ve test.....	56
Şekil 3.4. Senaryo 3: 60 hibrit özellik ile eğitim ve test .....	57
Şekil 3.5. Orijinal görüntünün optimal piksel yolu (üstte), dört şerit ve her şeridin optimal piksel yolu (altta).....	60
Şekil 3.6. Piksel yolu çıkarma tespiti için değerlendirilen şeritler.....	61
Şekil 3.7. Muhtemel tüm piksel yollarının kümülatif toplamlarını hesaplayan algoritma.....	62
Şekil 3.8. Şerit tabanlı yöntemin akış şeması.....	63
Şekil 4.1. UCID veritabanından örnek görüntüler .....	65
Şekil 4.2. KMEM'deki muhtemel piksel yolları .....	66
Şekil 4.3. %30 ölçeklemenin karışık veri kümesi ile testi .....	70
Şekil 4.4. Tüm yöntemlerin tespit doğruluklarının karşılaştırılması .....	72
Şekil 4.5. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%3 için) .....	73
Şekil 4.6. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%6 için) .....	74
Şekil 4.7. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%10 için) .....	74
Şekil 4.8. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%20 için) .....	75
Şekil 4.9. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%30 için) .....	75

## TABLolar LİSTESİ

Tablo 3.1. Enerji tabanlı özellikler.....	54
Tablo 3.2. Piksel yolu tabanlı özellikler .....	54
Tablo 3.3. Gürültü tabanlı özellikler.....	54
Tablo 3.4. Bir şeridin piksel yolu tabanlı özellikleri.....	59
Tablo 4.1. Hibrit yaklaşım ile tespit doğrulukları (%).....	67
Tablo 4.2. Hibrit yöntemin Yanlış Pozitif Oranı (%) .....	67
Tablo 4.3. Hibrit yöntemin Yanlış Negatif Oranı (%) .....	67
Tablo 4.4. Tespit doğrulukları (%).....	68
Tablo 4.5. Karışık veri kümesi ile tespit doğrulukları (%) .....	69
Tablo 4.6. Şerit sayısına göre tespit doğrulukları (%) .....	71
Tablo 4.7. 6 şerit ve karışık veri kümesi ile tespit doğrulukları (%).....	71
Tablo 4.8. 6 şerit ile tespit diğer yöntemlerle tespit doğrulukları (%) açısından karşılaştırılması .....	72
Tablo 4.9. Tüm yöntemlerin başarımları ve 6 şeritli yöntemin [39] ile farkı .....	72
Tablo 4.10. ROC analizi.....	76

## ÖZET

Anahtar kelimeler: piksel yolu çıkarma tespiti, içerik duyarlı boyutlama, yerel ikili örüntüler

Piksel yolu çıkarma (seam carving), günümüzde en çok uygulanan içeriğe duyarlı görüntü boyutlandırma yöntemlerinden biridir. Piksel yolu çıkarmanın sebep olduğu bozukluklar çok yüksek oranlarda ölçekleme yapılmadıkça insan gözü tarafından algılanamaz. Bu görsel başarının sebebi görüntüdeki piksellerin önem değerlerine göre değerlendiriliyor olmasıdır. Görüntünün optimal seam'ı, görüntü genelinde toplamda en az enerji (önem) değerine sahip piksel yoludur. Tek piksel genişliğindeki önemsiz bu piksel yolları birer azaltılarak her iterasyonda görüntünün genişliği ya da yüksekliği bir azaltılır. Anlamsal olarak önemli olan ön plan nesnelere mümkün olduğunca dokunulmaz. Görüntünün içeriğinin bu denli korunduğu bir ölçekleme yaklaşımı kötü niyetli olarak da kullanılabilirdiğinden, bu şekilde ölçeklenmiş görüntülerin tespiti büyük önem arz etmektedir. Piksel yolu çıkarma tabanlı ölçeklemenin tespiti diğer ölçekleme yöntemlerine göre oldukça zordur; çünkü görüntülerin geometrik açıdan ele alınması yetmez, anlamsal bir değerlendirme içeren detaylı bir analiz yapılması gerekmektedir.

Bu çalışmada, piksel yolu çıkarılarak boyutları değiştirilmiş görüntülerin tespiti, görüntülerden özellik çıkarılması ve çıkarılan özelliklerle Destek Vektör Makinesi'nin eğitilmesi şeklinde gerçekleştirilmektedir. Çıkarılan özellikler piksel yolu çıkarma algoritmasının uygulanışı ile alakalı özelliklerdir. Ayrıca, yöntemin başarımını artırmak amacıyla, özellik çıkarımı öncesinde görüntülere Yerel İkili Örüntüler dönüşümü uygulanmış ve piksel yolu çıkarmanın sebep olabileceği yerel bozukluklar belirginleştirilmiştir. Tüm bunlara ek olarak, piksel yolu çıkarmanın görüntülerin farklı parçalarındaki etkileri de incelenmiştir. Bu amaçla görüntüler şeritlere ayrılarak her bir şerit seam özellikleri bakımından değerlendirilmiş ve tespit doğrulukları bu şekilde oldukça artırılmıştır.

Geliştirilen yöntem ile piksel yolu çıkarma tabanlı ölçekleme %30 ölçeklenmiş görüntülerde %99,9'lara kadar tespit edilebilmiştir. Performans literatürdeki diğer yöntemlere göre ortalamada %20'den fazla artırılmıştır. Tespit performansı özellikle tespit edilmesi daha zor olan %3, %6 gibi küçük ölçekleme oranlarında %26 geliştirilmiştir.

# **DETECTION OF SEAM CARVING BASED SIZE MODIFICATION IN DIGITAL IMAGES**

## **SUMMARY**

Keywords: Seam carving detection, content-aware resizing, local binary patterns

Seam carving is one of the mostly applied content-aware image resizing methods today. The deteriorations caused by seam carving are mostly unnoticeable for human eyes unless the scaling ratio is very high. The reason of this visual success comes from evaluating the pixels according to their importance values. Optimal seam of an image is a pixel path which contains the least energy (importance) throughout the image. Image width or height is decreased by one in each iteration by removing those unimportant, one-pixel width pixel paths. The semantically important foreground objects remain untouched as far as possible. Since such a scaling approach which perfectly preserves the image content can be used malevolently, the detection of the images that are scaled in this manner becomes more of an issue. The detection of seam carving is more difficult than the other scaling methods since evaluating the images geometrically is not sufficient, but a detailed analysis investigating the semantical concept is required.

In this study, the detection of the images scaled by seam carving is realized by feature extraction and training a Support Vector Machine with those features. The extracted features are related to the seam carving process. In addition, Local Binary Patterns transform is applied to the images before feature extraction to reveal the local artifacts caused by seam carving. Besides, the effect of seam carving in sub parts of the images is investigated. For this purpose, the images are divided into several stripes and each and every stripe is evaluated in terms of seam features. This evaluation has been improved the detection accuracies.

Seam carving based resizing has been detected up to 99,9% in 30% scaled images by the developed method. The detection performance has been improved 20% on the average when compared with other methods in the literature. The detection performance is improved 26% in low scaling ratios like 3% and 6% which are harder to detect.

## **BÖLÜM 1. GİRİŞ**

Görüntülerin farklı boyutlara dönüştürülmesi pek çok görüntü işleme uygulaması için temel ve kullanışlı bir işlemdir. Görüntüyü yeniden boyutlandırırken, aynı zamanda içindeki ön plan nesnelere ve içeriğini korumak görüntü işleme alanında karşılaşılan en popüler araştırma konularından biridir. Kırpma ve ölçekleme gibi geleneksel boyutlandırma yöntemleri, veri kayıplarına (kırpma) ve içerikte gözle görülür bozukluklara (ölçekleme) sebep olduğu için yeterli değildirler. Bu sebeple, son zamanlarda araştırmacılar içeriğe duyarlı görüntü boyutlandırma yöntemi arayışı içindedirler. Bu amaçla, Avidan ve Shamir etkileyici bir yaklaşım önermişlerdir [1]. Önerileri görüntünün genişliğini/yüksekliğini azaltmak için en önemsiz piksel yollarını bir bir silmek, boyutu artırmak için ise önemsiz piksel yolu ile onun komşusu arasına interpolasyon ile oluşturulmuş yapay piksel yolu eklemek şeklindedir. Bu yaklaşım geleneksel ölçeklendirme yöntemlerinin pek çok dezavantajını ortadan kaldırarak daha iyi görsel kalite sağlar.

Dijital görüntü araçlarının yaygınlaşmasıyla görüntülerin özellikle de içerik duyarlı olarak boyutlandırılması, görüntülerin dinamik olarak boyutlandırıldığı akıllı telefonlarda, monitörlerde ve web’de daha çok kullanılmaktadır. Görüntüler klasik ölçekleme ile büyütülüp küçültülebilir ve hatta genişlik ve yükseklik farklı oranlarda değiştirilebilir. Fakat ölçekleme, görüntüdeki tüm nesnelere eşit olarak uygulandığından görüntüdeki bazı nesnelere önemini yitirebilir. Bu sebeple, içerik korunmadığından iyi bir yöntem değildir. Öte yandan, optimal piksel yolunu bularak görüntüyü büyütme ya da küçültme için yalnızca önemsiz piksel yolları eklenir ya da çıkarılır. Dolayısıyla, görüntüler içerikleri bozulmadan boyutlandırılır.

Seam, bir görüntüdeki soldan sağa ya da yukarıdan aşağıya giden, optimal, 8 bağlantılı ve optimalliğin enerji fonksiyonu ile belirlendiği bir piksel yoludur [1]. Piksel yolu çıkarmanın temel mantığı, görüntüyü istenen yönde büyütme için piksel yolları eklemek ve küçültmek için de piksel yolları çıkarmak şeklindedir. Sonuç olarak bu yöntem, görüntünün temel özelliklerini korurken, en göz ardı edilebilecek pikselleri de çıkarır. Her satırdan veya sütundan birer piksel çıkarılması görüntünün dikdörtgen yapısını korunmasını sağlamaktadır.

Piksel yolu çıkarma sırasında bazı hesaplamalar yapılması sebebiyle işlem hızı diğer yöntemlerle karşılaştırıldığında daha yavaştır. Kırpma, en hızlı görüntü boyutlandırma yöntemidir; çünkü kırpma penceresi dışında kalan alan hiçbir hesaplama yapılmaksızın doğrudan atılır. Ölçeklemede ise görüntüyü büyütme için komşu pikseller arasına, değeri komşuların ortalaması olan ara pikseller eklenir. Görüntüyü küçültürken de bazı pikseller birleştirilir (komşu iki piksel yerine değeri bu piksellerin ortalaması olan bir piksel yerleştirilir). Piksel yolu çıkarma bu iki yeniden boyutlandırma yöntemine göre oldukça karmaşıktır. Öncelikle bir enerji haritasının çıkarılmasını gerektirir ki bu da kullanılacak enerji fonksiyonuna bağlı olarak çok zaman alabilmektedir. Optimal seam çıkarıldıktan sonra diğer pikseller soldan sağa ya da yukarıdan aşağıya doğru kaydırılır. Performans algoritma içindeki bazı parametrelere göre bile değişebilir.

Piksel yolu çıkarma algoritması ölçeklemenin yanında değişik amaçlarla da kullanılmıştır. Bunlardan biri fotoğrafların estetik açıdan güzelleştirilmesidir [2]. Bir fotoğrafın estetik açıdan güzel olabilmesi için içindeki nesnelerin optimal pozisyonlarında olup güzel bir kompozisyon sunması gerekir [2]. Bu çalışmada, enerji fonksiyonu ile her bir pikselin çıkıntı (saliency) değeri ölçülmüş ve çıkarılacak ya da eklenecek pikseller bulunmuştur. Resimdeki nesnelerin kesilip önerilen yerlere yerleştirilmesi yerine piksel yolu çıkarma tabanlı, nesnelerin optimal pozisyonlara getirilmesine yönelik yeni bir yaklaşım önerilmiştir. Böylelikle resmin çözünürlüğü korunarak estetiği iyileştirilmiştir. Farklı bir amaçla, avuç izi tanımada [3] piksel yolu çıkarma kullanılmıştır. Önemsiz piksel yollarının bulunması yerine en önemli piksel yolları bulunmuş ve avuç izleri ile bulunan bu piksel yolları eşleştirilmiştir.

[4]'te piksel yolu çıkarma belgelerdeki metin satırlarının çıkarılmasında kullanılmıştır. Gri seviyeli ve ikili resimler için enerji haritası çıkarımında sırasıyla gri uzaklık ve işaretli uzaklık dönüşümleri kullanılmıştır. Her bir satır için orta piksel yolları ve ayırıcı piksel yolları hesaplanmıştır. Orta piksel yolları varsayılan metinlerin üzerinden geçerken, ayırıcı piksel yolları metin satırlarının alt ve üst sınırlarından geçmektedir. Aşağıda piksel yolu çıkarma ile alakalı mevcut çalışmalar üç gruba ayrılarak incelenmiştir.

### 1.1. Görsel Kalitenin İyileştirilmesi

[5]'de önem yayılımı kullanarak piksel yolu çıkarma görsel açıdan iyileştirilmiştir. Ardışık olarak çıkarılan önemsiz piksel yolları görüntüde çoğu zaman gerdirme etkisi oluşturabilmektedir. Bu ve benzeri görsel bozuklukları giderebilmek için çıkarılan seam'deki her bir pikselin önem değeri komşu piksellerine yayılmıştır. Böylelikle, görüntüdeki gerdirme ve süreksizlik bozuklukları indirgenmiştir. Başka bir çalışmada, görüntüleri boyutlandırırken tek bir operatör kullanmanın sebep olduğu görsel bozukluklar, birden fazla operatör kullanarak azaltılmaya çalışılmıştır [6]. Piksel yolu çıkarma, kırpma ve ölçekleme ile birlikte kullanılarak görsel olarak iyileştirilmiştir. Yapılan kullanıcı (bilgisayar bilimi öğrencileri ve grafik tasarımcılar) analizleri neticesinde görüntüyü boyutlandırmada tek bir operatör kullanmak yerine birçok operatörün birlikte kullanılmasının daha iyi sonuçlar ürettiği gösterilmiştir. Tüm operatörler sıralı bir dizide birleştirilmiştir. Operatörlerin farklı oranlardaki kombinasyonları ile deneyler gerçekleştirilmiştir. Operatörlerin kullanım sıralarından çok kullanım oranlarının önemli olduğu varsayılmıştır. Bu da, problemin dinamik programlama ile formüle edilmesine yönelmiştir. Operatörlerin maliyeti hedef görüntü ile orijinal görüntü arasındaki benzerlik olarak hesaplanmıştır. İdeal sıralı dizi, maliyete bağlı tüm sıralı dizilerin olasılıklarının hesaplandıktan sonra minimum maliyete sahip olan operatör dizisidir.

Çok operatörlü yöntemlerin hem yatay hem dikey boyutlandırma hesaba katıldığında sebep olabileceği büzme ve gerdirme etkileri, doğrudan ve dolaylı piksel yolu çıkarma birleştirilerek giderilmeye çalışılmıştır [7]. Önerilen çok operatörlü yöntem,



operatör deęişimine karar verme maliyetini yeni tanımlanan görüntü bozukluk ölçütü sayesinde önemli ölçüde azaltmıştır. Ayrıca görüntünün genel yapısını korumak için piksel yolu çıkarmaya yeni bir iyileştirme yapılmıştır. Görüntünün genişliği azaltılırken öncelikle görüntüde bozukluk oluşana kadar piksel yolu çıkarma uygulanıyor (doğrudan piksel yolu çıkarma). Sonra, görüntüye dikey piksel yolları ekleniyor (dolaylı piksel yolu çıkarma) ve bu ekleme bozukluklar oluşmaya başlayana kadar devam ediyor (bozuklukların derecesi önerilen bozukluk ölçütü ile belirleniyor). Daha sonra görüntü başlangıç yüksekliğine ölçekleniyor ve son olarak çıkıntı (saliency) tabanlı kırpma uygulanır. Piksel yolu çıkarmanın sebep olduğu görsel bozuklukları gidermek için görüntü yayılımı piksel yolu çıkarma ile birleştirilmiştir [8]. Kullanılan görüntü yayılımı yöntemi, kenar koruma tabanlı adaptif ICC (Interveining Contour and Color - Ara Sınır ve Renk) [9] yöntemidir. Önce bu teknikle görüntüdeki dağınık doku yumuşatılmış, sonra da piksel yolu çıkarma uygulanmıştır. Ara sınır ve renk bileşenlerini birleştirerek görüntülerin yapısı korunurken görüntüdeki dağınık doku da yumuşatılmıştır.

Piksel yolu çıkarma yönteminde karşılaşılan bozukluklar genellikle kullanılan enerji fonksiyonu ile alakalıdır; çünkü tüm görüntü türleri için mükemmel sonuç üretebilecek kapsamlı bir enerji fonksiyonu yoktur. [10]'da farklı kenar filtrelerinin görsel performansa etkisi incelenmiştir. Canny filtresinin daha iyi sonuçlar ürettiği gösterilmiştir. Başka bir çalışmada, yeni bir adaptif çıkıntı (saliency) modeli önerilmiş ve gradyanın  $L_1$  normu ile birleştirilerek enerji haritası elde edilmiştir [11]. Geleneksel yöntemlerin aksine görüntüleri iki boyut özellikleri değil üç boyut özellikleri değerlendirilmiştir. Gradyanın  $L_1$  normunun ve görüntü tanımındaki çıkıntının farklı karakteristik özellikleri, piksel yolu çıkarma performansını diğer gözde yöntemlere göre artırmıştır. Enerji fonksiyonu ile ilgili başka bir çalışmada düşük mertebeli gösterimin matris çözümlemesi yapılmıştır [12]. Nesnelerin sınır bilgisini yansıtan özellik matrisi düşük mertebeli ve seyrek matrisler olarak ayrılmıştır. Seyrek matrisin ROI (Region Of Interest-İlgi Bölgesi) bilgisini yansıttığı düşünülerek enerji fonksiyonu olarak kullanılmıştır. 1000 görüntü üzerinde yapılan deneyler, yöntemin etkili ve güçlü olduğunu göstermiştir.

[13]'de yeni bir çıkıntı (saliency) tespit yöntemi önerilmiş ve içeriğe duyarlı olarak görüntülerin boyutlandırılması için yeni bir seam kriteri sunulmuştur. Önerilen çıkıntı tespit yöntemi güvenilir bir görüntü önem haritası üretir ve bu harita görüntüdeki çıkıntı nesnelere piksel yolu çıkarılmasından korur. Yani, o bölgelerden daha az seam geçer. Önerilen yeni seam kriteri ise piksel yollarını görüntü üzerinde yaymak amaçlıdır. Böylece, çizgi ya da eğri şeklindeki yapılar korunmuş olur. Yöntemin karışık arka plana sahip görüntülerde bile çıkıntıları doğru tespit ettiği ve piksel yolu çıkarma sonrası büyük ölçekli geometrik yapılarda oluşan bozuklukların azaltıldığı gösterilmiştir. [14]'da piksel yolu çıkarma için yalnızca kenarların tespit edilmesi yerine, görüntüdeki önemli nesnelere sınırları çizilmek suretiyle tespit edilmiştir. Tespit işleminde seviye kümesi tabanlı Chan Vese Görüntü Bölütleme algoritması [15] ve sabit yakınsama oranlı Delta Bar Delta öğrenme algoritması kullanılmıştır. Bu şekilde önemli nesnelere çıkarıldıktan sonra piksel yolu çıkarma (seam carving) uygulanmış ve görsel açıdan daha iyi sonuçlar elde edilmiştir.

Pek çok içeriğe duyarlı görüntü boyutlandırma yaklaşımının aksine [16] danışmanlı öğrenmeye dayalı bir yöntem geliştirmiştir. Önceki bahsedilen çalışmalarda olduğu gibi piksel yolu çıkarma, görüntüdeki önemli sınırlara karar verdikten sonra uygulanmıştır. Böylece, önemli nesnelere üzerinden geçen seam sayısının en aza indirilmesi amaçlanmıştır. ROI çalışma kümesindeki görüntülerden öğrenilir. Daha sonra girdi resminin bu sınırları, hedef çıktı resmi elde etmek için piksel yolu çıkarma uygulanırken anahtar olarak kullanılmıştır. Bir başka çalışmada piksel yolu çıkarmanın sebep olduğu bozukluklar kullanılan filtrelerle azaltılmaya çalışılmıştır [17]. [18]'de kenarların korunumu artırılmış ve kırılma görünümü veren bozukluklar azaltılmıştır. Önerilen yaklaşımda, yerel gradyan bilgisi bir eşikleme tekniği ile birlikte kullanılmış ve seam seçimine rehberlik etmiştir. Ayrıca bu yöntemle kabul edilemeyecek görsel bozukluklar olduğu anda piksel yolu çıkarmayı durduracak bir mekanizma da mevcuttur. Kırılma bozuklukları, kırılma karşıtı filtrelerle giderilmiştir.

[19]'de zikzak bozukluklarını bastırmak üzere yeni bir enerji tanımı getirilmiştir. Enerji haritası Gabor filtresi ve çıkıntı (saliency) haritası kullanılarak

şekillendirilmiştir. Enerji hesabı ise bağlantı kısıtları olmaksızın (geleneksel piksel yolları 8 bağlantılı piksel yollarıdır) boyutlandırma yapabilmek için Gabor özellik uzayında optimize edilir. Bağlantılı piksel yollarının sebep olabileceği şekil bozuklukları böylelikle engellenmiş olur. [20]'de çiftli (stereo) görüntülerin piksel yolu çıkarma ile boyutlandırılması ele alınmıştır. Görüntülerin tek tek ele alınarak ayrı ayrı piksel yolu çıkarma uygulanması geometrik yapıyı bozmaktadır ve sahnenin üç boyutlu algılanmasını imkânsızlaştırmaktadır. Bunu çözmek için görüntü çiftlerindeki pikseller arasındaki görünürlük ilişkileri hesaba katılmıştır. Yöntem çok sayıda kapalı mekân ve dış mekân stereo görüntü üzerinde test edilmiştir. Bir başka görsel iyileştirme çalışmasında dalgacık analizinden faydalanılmıştır [21]. İnsan görsel sistemi ile uyumlu olması sebebiyle enerji haritasının çıkarımında dalgacık analizi kullanılmıştır. Görüntüler yeni boyutlarına dönüştürülürken anlamsal bütünlük korunarak görsel açıdan ölçekleme ya da klasik içeriğe duyarlı boyutlandırma yöntemlere göre daha iyi sonuçlar elde edilmiştir. [22]'de piksel yolu çıkarma yeni bir önem tespit yöntemi ile uygulanmıştır. Bu yöntemde kenar bilgisi ve görsel çıkıntı hesaba katılmıştır. Öncelikle görüntünün sıkıştırılabilirliği değerlendirilmiş ve her iki yön için uygun seam sayısı atanmıştır. Son olarak görüntünün hedeflenen boyutlara getirilmesi için klasik ölçekleme yapılmış ve daha iyi görsel sonuçlar elde edilmiştir.

## 1.2. İşlem Hızının Artırılması

[23]'de piksel yolu çıkarma algoritmasında, hesaplama yoğunluğunun kullanılan enerji haritasına bağlı olduğu gösterilmiştir. Görsel kalitenin iyileştiği kenar belirleme yöntemleri kullanıldığında hesaplama süresinin arttığı gösterilmiştir.

Piksel yolu çıkarma [24]'de işlem verimi açısından dalgacık ağacı ile birlikte uygulanmıştır. Önemsiz dalgacık ağaçlarının piksel yollarını ölçek özyinelemeli olarak adım adım çıkarmak için dalgacık ağaçlarının hiyerarşik yapısından faydalanılır. Önerilen algoritmanın, işlenen resmin boyutu büyüdükçe daha avantajlı olduğu vurgulanmıştır. Çıkıntı (saliency) özellikleri olarak dalgacık katsayılarını

kullanması sebebiyle görsel açıdan da klasik piksel yolu çıkarma algoritmasına göre daha iyi olduğu örneklerle gösterilmiştir.

[25]'de piksel yolu çıkarmanın karmaşık hesaplamaları ve hafıza erişimleri ile sebep olduğu vakit kayıplarını gidermek için yeni bir kümülatif olmayan piksel yolu çıkarma yöntemi önerilmiştir. Piksel yolu çıkarma algoritmasında, her bir pikselin minimum kümülatif enerji matrisindeki değeri dinamik programlama ile hesaplandığı için, bir pikselin kümülatif enerji değerini hesaplanırken pikselin üst komşularının kümülatif değerlerinin hesaplanmış olmasını gerekir. Bu durumda paralel olarak iş yapmak zorlaşır. Bunun için, her pikselin kümülatif değeri değil de kendi değeri kullanılarak enerji haritası çıkarılır ve seçilen pikselin indeksi indeks haritasına kaydedilir. Bu işlem her piksel için eşzamanlı olarak yapılabilir. Dolayısıyla, paralelleşmeye müsait hale gelmiş olur. Daha sonra bu indeks haritasından seam elde edilir. İşlem süresi bu yöntemle tek GPU kullanılarak on kata kadar iyileştirilmiştir. Ayrıca, orijinal piksel yolu çıkarmanın tek iş parçacığı kullanan CPU uygulamasına göre maksimum da 103 kat hızlanma iki GPU'lu sistemle sağlanmıştır.

[17]'da her seferinde bir tane seam çıkarmak yerine, işlem hızını artırmak amacıyla fazla sayıda seam aynı anda çıkarılmıştır. Bunun için daha büyük filtreler kullanılmıştır ve işlem süresi azaltılmıştır.

### **1.3. Piksel Yolu Çıkarma Tabanlı Ölçeklemenin Tespiti**

Piksel yolu çıkarma algoritması kullanılarak yapılan ölçekleme işlemlerinin tespiti ilk olarak algoritmanın önerilmesinden iki yıl sonra başlamıştır. Sarkar ve diğerleri [26], piksel yolu çıkarmayı sahtekârlık tespiti açısından inceleyen ilk araştırmacılarıdır. Makine öğrenmesi tabanlı bir tespit yöntemi geliştirmişlerdir. Bir görüntünün orijinal mi piksel yolu çıkarma ile ölçeklenmiş mi olduğunun tespiti, 324 boyutlu Markov özelliği ve AKD (Ayrık Kosinüs Dönüşümü) domenindeki 2 boyutlu histogramlar aracılığı ile gerçekleştirilmiştir. Çıkarılan özellikler ile DVM (Destek Vektör Makinesi)'de eğitim yapılmıştır. Piksel yolu çıkarma ve seam ekleme sırasıyla %80 ve %85 tespit edilmiştir. Seam eklemenin tespitinde her yeni pikselin,

komşularının lineer bir kombinasyonu olduğu varsayılmıştır ve düşük ekleme oranlarında yüksek tespit oranları (%94) elde edilmiştir. [27]'de bir görüntü tanıma yaklaşımı önerilmiştir. Çalışmada DVM, dört grup özellik ile eğitilmiştir. Tüm özellik vektörleri görüntünün gradyan histogramı üzerinden hesaplanmıştır. Özellikler enerji tabanlı, seam davranışı tabanlı, yüksek dereceli istatistik tabanlı ve ikinci piksel yolu çıkarma tabanlı olmak üzere gruplanmıştır. Enerji tabanlı özelliklerin ilki, belirlenen enerji eşiği altında kalan piksellerin yüzdesinden oluşmaktadır. İkinci ve üçüncüsü ise sırasıyla orta seviye ve yüksek seviye histogram değerlerinden oluşmaktadır. Sonuncusu ise ADF (Ağırlıklı Dağıtım Fonksiyonu)'den elde edilmektedir. Daha iyi ayırım için normalize edilmiş ADF kullanılmıştır. Seam davranışı tabanlı özellikler, belli noktalara kadar bulunan piksel yolları arasındaki mesafe ve buralardaki ağırlıklı enerjiden oluşmaktadır. Çalışmadaki en önemli çıkarım şudur: piksel yolu çıkarma uygulanmamış bir görüntüde belli noktalara kadar bulunan piksel yolları sürekli bir hat şeklinde iken, piksel yolu çıkarma uygulananlarda bu sürekliliğin bozulması, piksel yollarının dağılmasıdır. Araştırmacılar ayrıca, piksel yolu çıkarmanın görüntülerin istatistiğini etkilediğini iddia etmişler ve wavelet absolute moment özelliklerini de kullanmışlardır. Son olarak, piksel yolu çıkarmayı eldeki şüpheli görüntüye tekrar uygulamışlar ve bahsedilen tüm özellikleri bu resim üzerinden çıkarmışlardır. Yapılan deneyler sonucunda, yöntemin görüntülerin yeniden boyutlandırılması konusunda başarılı olduğu; fakat nesne çıkarma konusunda aynı başarıyı gösteremediği görülmüştür.

Bir başka çalışmada piksel yolu çıkarmanın tespiti görüntülere eklenen kenar bilgisi yardımıyla gerçekleştirilmiştir [28]. Eklenen bilgi SIFT (Scale Invariant Feature Transform - Ölçeklemeden Bağımsız Özellik Dönüşümü)'in görsel kelimeler gösterimi tabanlıdır. Bu metot ile yalnızca piksel yolu çıkarma uygulanıp uygulanmadığı değil, nerelere ve hangi miktarda uygulandığı da belirlenebilmektedir. Orijinal ve işlenmiş görüntülerdeki ardışık SIFT noktaları arasındaki mesafeler arasındaki fark hesaplanmaktadır. Pozitif fark o noktadan seam çıkarıldığını, negatif fark ise seam eklendiğini göstermektedir. Yöntemde çok yüksek tespit oranları (ortalama %99.4) elde edilse de orijinalden bağımsız (blind) bir tespit

yöntemi olmayıp, orijinale ait bilgi gerektirdiğinden tercih edilebilecek bir yöntem değildir.

[29]'de piksel yolu çıkarma işlemi, piksel yolu çıkarma algoritmasının özellikleri kullanılarak tespit edilmeye çalışılmıştır. Piksel yolu çıkarma algoritması düşük enerjili piksel yollarını görüntüden çıkarır ve kalan görüntünün ortalama enerjisi artmış olur. Bu amaçla çalışmada, enerjiyle alakalı on dört istatistikî özellik çıkarılmıştır. Çıkarılan özelliklerle DVM'de eğitim ve test yapılmıştır. Seam eklemenin tespiti için komşu pikseller arasındaki korelasyon kullanılmıştır. UCID veri tabanından [30] alınan orijinal resimler ve bunların belli oranlarda piksel yolu çıkarma ile ölçeklenmiş halleri ile veri tabanı oluşturulmuştur. Toplamda 8028 resimden oluşan veri tabanı ile yapılan deneylerde piksel yolu çıkarma ile genişliği %50 azaltılmış resimler %93.50 tespit edilebilmiştir.

Piksel yolu çıkarma ile ölçeklemenin tespitinde farklı bir yaklaşım [31]'da verilen poster çalışmasında önerilmiştir. Görüntüler üzerinde yama analizi yapılmıştır. Görüntü  $2 \times 2$ 'lik alt görüntülere (mini kareler) ayrılmış ve bu mini karelerle uyumlu dokuz yama oluşturulmuştur. Ayrıca bir de kosinüs benzerliği ile uygun yamaya karar veren hakem örüntü kullanılmıştır. Yamalar komşular arasına interpolasyon ile oluşturulmuş pikseller eklenerek oluşturulmuştur. Yamalar oluşturulduktan sonra, mini kareleri dikey, çapraz ve ters çapraz yönlerde bağlayan yama geçiş olasılığı matrisleri oluşturulmuştur. Dokuz yamanın olasılıklarından ve geçiş olasılığı matrislerinin girdilerinden oluşan 252 boyutlu özellik vektörü DVM'nin eğitim ve testinde kullanılmıştır. %20, %30 ve %50 ölçeklenmiş görüntülerin tespit oranları sırasıyla %92.2, %92.6 ve %95.8 olarak bulunmuştur. [29]'deki yöntemde görüntünün gürültüsü ile alakalı dört yeni özellik eklenmiş ve tespit oranını artırmak amaçlanmıştır [32]. Piksel yolu çıkarma tabanlı görüntü ölçeklemenin, görüntünün gürültü seviyesini etkilediği çıkarımından yola çıkarak, Wiener filtresi kullanılarak elde edilen gürültü matrisinin ortalama, standart sapma, çarpıklık ve basıklık değerleri hesaplanmıştır. Wei ve diğerlerinin poster çalışması [33]'de genişletilmiştir. Yama analizinin ayrıntıları verilmiş ve ileri yönlü ve geri yönlü piksel yolu çıkarmanın sonuçları karşılaştırılmıştır. İleri yönlü piksel yolu

çıkarmasının daha kolay tespit edilebildiği gösterilmiştir. Sebebi ise piksel yollarının daha geniş yayılmasına bağlanmıştır. Çalışmada ayrıca yamaların geçiş olasılık matrisleri de değerlendirilmiş ve mini karelerin optimal yamalarının 0, 4 ve 8. türlerde yoğunlaştığı gösterilmiştir. Benzerlik ölçütü olarak Öklit mesafesi de denenmiş ve resimdeki sıcak bölgeler analiz edilmiştir. Mini karelere yamaların geçiş olasılıklarına bağlı olarak skor verilmiş ve düşük skorlu mini kareler göz ardı edilmiştir. Yüksek skora sahip olan mini kareler görüntüdeki sıcak bölgeleri ifade etmektedir. [34]'de yama analizi piksel yolu eklemenin tespitinde kullanılmıştır. Aynı yöntem izlenmiş; fakat kosinüs benzerliği yerine yamaların hakem yama ile karşılaştırılmasında Öklit mesafesi kullanılmıştır. % 94 ve üzeri tespit oranları elde edilmiştir.

Bir diğer orijinalden bağımsız (blind) piksel yolu çıkarma tespit yönteminde BACM (Blocking Artifact Characteristics Matrix)'den istifade edilmiştir [35]. BACM düzenli simetrik şekilleri ortaya çıkarır. Bozulan (ölçeklenen) görüntülerde BACM'nin düzenli simetriklik özelliği yok olur. JPEG görüntülerden BACM elde edildikten sonra on sekiz özellik çıkarılır ve DVM'nin eğitiminde kullanılır. %50 küçültülen görüntüler %93.24 oranında tespit edilirken, %50 büyütülen görüntüler %94.77 tespit edilmiştir. Videolardaki piksel yolu çıkarma tabanlı boyutlamaların tespiti için [36]'de adli sağlama (forensics hash) tabanlı bozulma tespiti ve konumlama yaklaşımı önerilmiştir. Yöntem her zaman-uzamsal görüntüden sabit Hızlandırılmış Sağlam Özellik (Speeded-up Robust Feature-SURF) noktaları çıkarır ve eşleşme yüzeylerini gösterir. Komşu eşleşme yüzeylerinin pozisyon değişimleri, anahtarın (hash) bütünleşik ve ölçeklenebilir bir şekilde oluşturulmasında kullanılır. Yöntemde çıkarılan piksel yollarının miktarı net olarak belirlenebilirken, konumu da kabaca belirlenmiştir. Bu belirlemede anahtarın (hash) uzunluğu önemli rol oynamıştır. Çalışmada, SIFT yerine SURF noktalarının çıkarılmasının sebebi büyük ölçüde işlem hızının etkin kullanımına bağlanmaktadır. SURF'un işlem hızı ve sağlamlık bakımından SIFT'e göre üstünlüğü [37]'de gösterilmiştir. Videolar üzerinde işlem yapılacağından, fazla miktarda veri hesaba katılacaktır. Bu sebeple işlem hızı oldukça önemlidir. Ayrıca, döndürme ve ışıklandırma türündeki değişiklikler piksel yolu çıkarma tespitinde düşünülmesi gereken konulardan

değildir. Bu sebeplerle SIFT noktaları yerine SURF noktaları kullanmanın piksel yolu çıkarma tabanlı video boyutlandırmanın tespiti için daha uygun olacağına karar verilmiştir. Yapılan deneyler sonucunda ortalama %90 üzerinde tespit oranları elde edilmiş; fakat yüksek ölçekleme oranları ile denemeler gerçekleştirilmiştir. Örneğin, videoda %10 ölçekleme yapılması durumunda tespit oranının ne olacağı değerlendirilmemiştir. %70, %80 ve %90 ölçekleme oranları için yöntem değerlendirilmiştir. Bunun yanında yine [28] ile aynı sebeple yöntem kullanışlı değildir, orijinal veriye ait bilgi gerektirmektedir.

BACM tabanlı piksel yolu çıkarma tespit yaklaşımı [38]'da detaylandırılmıştır. BACM üzerinden yirmi iki özellik çıkarılarak matristeki simetri tanımlanmıştır. Bu özellikler DVM ile eğitilerek bir sınıflandırma modeli oluşturulmuştur. Görüntü önce  $8 \times 8$  bloklara sonra her blok kendi arasında yedi bölgeye ayrılarak bu bölgelerin simetrisi ilk sekiz özellikte değerlendirilmiştir. Daha sonra merkez bölge pikselinin diğer altı bölgede olma olasılığı altı yeni özellik eklemektedir. Ayrıca, yatay, dikey, çapraz ve ters çapraz yönlerde fark dizileri hesaplanmış ve ortalama değerleri özellik vektörüne katılmıştır. Son olarak, blok sınırlarının bozulması ile alakalı dört özellik eklenmiş ve toplamda 22 özellik DVM'ye verilmiştir. Yapılan deneylerde farklı kalite faktörleri (QF) ile JPEG formatında sıkıştırılan görüntüler kullanılmaktadır. Sıkıştırmada kalite faktörüne bağlı olarak tespit oranlarının önemli ölçüde değiştiği gösterilmiştir. En başarılı tespit QF100 ile sıkıştırılan görüntülerden elde edilmiştir; fakat özellik çıkarımı öncesinde veri tabanındaki tüm görüntülerin QF100 ile sıkıştırılması zaman karmaşasını oldukça artırmaktadır. Bir başka çalışmada, piksel yolu çıkarmanın tespiti tekrarlanan piksel yolu çıkarma işleminin ardından görüntünün önceki hali ile piksel yolu çıkarma sonrasındaki halinin benzerlik, enerji ve optimal piksel yolları arasındaki mesafe açısından değerlendirilmesi ile gerçekleştirilmektedir. Bahsedilen üç grup özellik DVM ile eğitilmektedir. Bu yöntem ile piksel yolu çıkarma ile %50 ölçeklenmiş görüntüler %98.85 oranında tespit edilebilmiştir.

Bahsedilen çalışmalardan farklı olarak, piksel yolu çıkarma sonucu oluşan bozuklukları belirginleştirmesi amacıyla [39]'de bir görsel tanımlayıcı kullanılmıştır.



Eldeki tüm görüntüler Yerel İkili Örüntüler (YİÖ) domenine dönüştürülerek görüntüdeki yerel bozukluklar belirginleştirilmiştir. Piksel yolu çıkarma algoritması, optimal piksel yolunun belirlenmesi ve bir bir görüntüden bu piksel yolları çıkarılarak kalan piksellerin kaydırılması şeklinde çalıştığından, görüntüde olabilecek bozukluklar çıkarılan yada eklenen piksel yoluna komşu piksellerde meydana gelmektedir. Yani bozukluklar görüntünün tamamında değil yerel bölgelerinde olmaktadır. Bu sebeple, YİÖ tabanlı piksel yolu çıkarma tespit yaklaşımı oldukça başarılı tespit oranları vermektedir. Yöntemde YİÖ dönüşümü yapılan görüntülerden [32]'daki on sekiz istatistiksel özellik çıkarılmıştır. Ayrıca, resmin yarısına ait bilgi içeren altı yeni özellik de eklenmiştir. Bu yeni özelliklerin kullanılmasındaki mantık şudur: görüntünün tamamı baz alınarak bulunan optimal piksel yolu, resmin yarısı baz alınarak bulunan optimal piksel yolundan farklıdır. Dolayısıyla, genel optimal piksel yolu resmin üst kısmı için optimal değildir ve bu piksel yolunun görüntüden çıkarılması, o bölgede bozukluklara sebep olur. Toplam yirmi dört özellik ile eğitilen DVM ile farklı ölçekleme oranları için testler yapılmıştır. Deneyler sonucunda mevcut diğer tespit yöntemlerine göre çok daha başarılı tespit oranları elde edilmiştir. Bu yöntem hem dikey hem yatay tespit için başarılı sonuçlar üretmektedir [40]. Görüntü boyutu büyüdüğünde başarımda düşüş yaşanmaktadır [41].

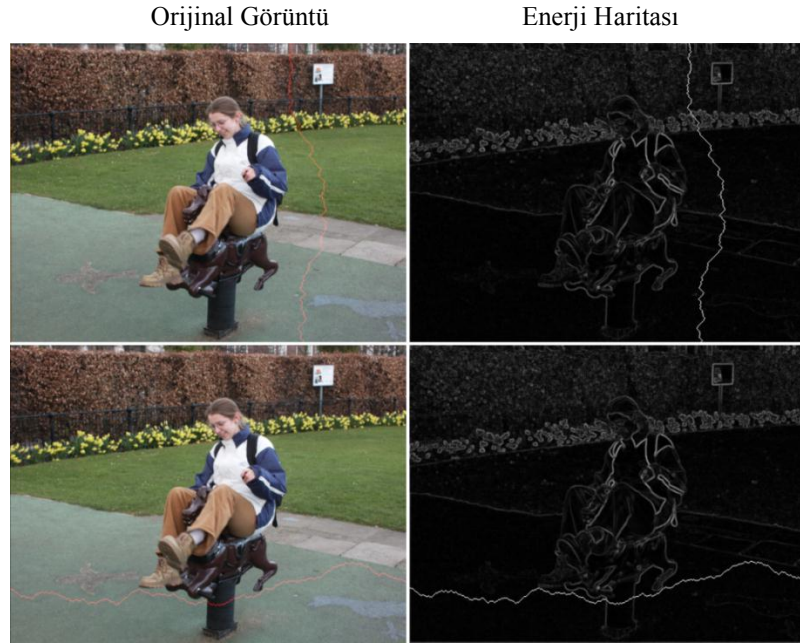
Bu tez çalışmasında piksel yolu çıkarma tabanlı içeriğe duyarlı ölçeklemenin önerilen şerit tabanlı yaklaşımla tespit edilmesi amaçlanmaktadır. Literatüre yapılan katkılar şöyle özetlenebilir:

1. Tespitte en etkili özelliklerin (piksel yolu tabanlı) kullanılması
2. Görüntünün farklı bölümlerinde piksel yolu çıkarmanın etkisinin incelenmesi
3. Optimal piksel yolu yerine muhtemel tüm piksel yollarının hesaba katılmasıyla daha kararlı, kullanılan piksel yolu seçim yönteminden bağımsız özellik seçimi
4. Tespit doğruluğunun en yüksek olduğu şeritlerin belirlenmesi
5. Şeritlere ayırmadaki optimum derinliğin belirlenmesi
6. GPU (Graphics Processing Unit) ile hızlandırılmış parametre seçimi

## BÖLÜM 2. PİKSEL YOLU ÇIKARMA (SEAM CARVING) VE SINIFLAYICILAR

Bu bölümde, tez çalışmasında yürütülen uygulamaları geliştirirken bilinmesi gereken ön bilgileri verilecektir. Piksel yolu çıkarma tabanlı görüntü boyutlandırmanın tespiti için öncelikle piksel yolu çıkarma algoritmasının anlaşılması gerekmektedir. Ayrıca, tespit yaklaşımında kullanılacak, yerel doku bozukluklarını belirginleştiren Yerel İkili Örüntüler yöntemi ve veri seti ile eğitim ve test yaparak sınıflandırmayı yapacak Destek Vektör Makineleri (DVM), DVM'nin uygun parametrelerinin belirlendiği ızgara arama yöntemi, DVM'ye verilecek eğitim ve test girdilerinin ayrıldığı dört katlı çapraz sağlama yöntemi, geliştirilen yöntemin önceki yöntemlerle karşılaştırılmasında kullanılacak analiz yöntemi olan ROC eğrileri önceden bilinmelidir. Bu amaçla, önce bu temel beş konu açıklanacaktır.

### 2.1. Piksel Yolu Çıkarma (Seam Carving)



Şekil 2.1. Orijinal görüntü ve enerji haritası üzerinde yatay ve dikey piksel yolları

Piksel yolu (seam), bir görüntüdeki soldan sağa ya da yukarıdan aşağıya giden, optimal, 8 bağlantılı ve optimalliğin enerji fonksiyonu ile belirlendiği bir piksel yoludur [1]. Enerji fonksiyonları görüntüdeki piksellerin önemini tanımlar. Ardışık olarak piksel yolu çıkararak ya da ekleyerek görüntüler istenen boyutlara içerikleri bozulmadan getirilir (Şekil 2.1.). Görüntünün boyutu azaltılırken piksel yolu seçimi, görüntünün yapısını koruyarak, daha çok düşük enerjili piksel ve daha az yüksek enerjili piksel çıkarılmasını garantiler. Görüntü büyütülürken de eklenecek piksel yollarının sırası orijinal görüntü içeriği ile yapay olarak eklenen pikseller arasındaki dengeyi korur. Dolayısıyla, bu operatörler, piksel yolu ekleme ve çıkarma operatörleri, içerik duyarlı olarak ölçeklenmiş görüntüler üretir.

Piksel yolu çıkarmada amaç, etrafıyla karışan, fark edilmeyen pikselleri çıkarmaktır [1]. Bu amaçla makale genelinde I görüntüsü için Denklem 2.1’de verilen basit enerji fonksiyonu kullanılmıştır. Fakat kullanılacak muhtemel enerji fonksiyonlarından da bahsedilmiştir. Örneğin, gradyanın  $L_1$  ve  $L_2$  normu, çıkıntı ölçütü [42], Harris köşeler ölçütü [43], göz hareketleri ölçütü [44] ve yüz algılayıcının çıktılarını enerji fonksiyonu olarak kullanılabilir.

$$f_{en}(I) = \left| \frac{\partial I}{\partial x} \right| + \left| \frac{\partial I}{\partial y} \right| \quad (2.1)$$

Görüntünün dikdörtgen yapısını bozmamak adına piksel yolu çıkarma algoritması hedeflenen boyutlara gelene kadar her satır ya da sütundan birer piksel çıkarır/ekler. Her satırdaki minimum enerjiye sahip belli sayıdaki pikseli çıkarmak enerjinin korunumu ilkesine uyarken görüntüde zikzak etkisi oluşturmaktadır. Görüntünün sütunlarının enerji bakımından değerlendirilerek en az enerjiye sahip belli sayıdaki sütunun çıkarılması ise yine görsel bozukluklara ve içerik kayıplarına sebep olmaktadır. Hedef ölçüye göre görüntünün kırılması da düşünülebilir; fakat bunda da görsel bozukluk olmasa da kırılma penceresi dışında kalan alan atılacağından içerik korunmamış olur. Klasik ölçeklemede ise görüntüdeki tüm nesnelere eşit oranda ölçekleme yapıldığından önemli ön plan nesnelere önemini kaybedebilmektedir. Şekil 2.4.’de tüm bu görüntü boyutlama yöntemlerinin etkileri

örnek görüntülerle gösterilmiştir. Şekilde orijinal görüntülerin genişlikleri beş farklı yöntemle %30 azaltılmıştır.

---

**Algorithm 1** Seam Carving-Genişlik Azaltma Algoritması

---

**Input:**  $I(n \times m)$ ,  $k$ (çıkarılacak seam sayısı)

**Output:** Boyutlandırılmış görüntü  $I'$

```

1:  $l \leftarrow$  Giriş görüntüsünü oku
2: for  $i \leftarrow 1, k$  do
3:    $E \leftarrow$  Enerjiyi Hesapla( $I$ )
4:    $M \leftarrow$  MinKumulatifEnerji( $E$ )
5:    $M \leftarrow E$ 
6:   for  $p \leftarrow 2, n$  do
7:     for  $q \leftarrow 1, m$  do
8:        $min_E \leftarrow \min(E_{p-1,q-1}, E_{p-1,q}, E_{p-1,q+1})$ 
9:        $E(p, q) \leftarrow E(p, q) + min_E$ 
10:    end for
11:  end for
12:   $seam(n) \leftarrow \min(E(n, :))$ 
13:  for  $j \leftarrow n - 1, 1$  do
14:     $seam(j) \leftarrow \min(E_{solust}, E_{ust}, E_{sagust})$ 
15:  end for
16:   $d \leftarrow I$ 'dan  $seam(n \times 1)$ 'i çıkar
17:   $d \leftarrow$  seam'den sonraki pikselleri sola kaydır
18:   $l \leftarrow d$ 
19: end for
20:  $l' \leftarrow l$ 

```

---

Şekil 2.2. Piksel yolu çıkarma algoritması (genişlik azaltma)

Tüm bu bahsedilen kısıtları gözetererek Avidan ve Shamir [1] piksel yolu çıkarmayı (seam carving) geliştirmişlerdir. Her satırdan ya da sütundan bir piksel çıkarılır fakat çıkarılan pikseller birbirine komşudur. Çıkarılan piksellerin yerlerini doldurmak için diğer pikseller sağdan sola ya da aşağıdan yukarıya kaydırılır. Böylece, piksellerin çıkarılmasının sadece yerel bir etkisi olur. Algoritmanın kaba kodu Şekil 2.2.'de verilmiştir.  $n \times m$  boyutlarındaki bir görüntü için dikey ve yatay piksel yolları Denklem 2.2 ve 2.3'de tanımlanmışlardır.

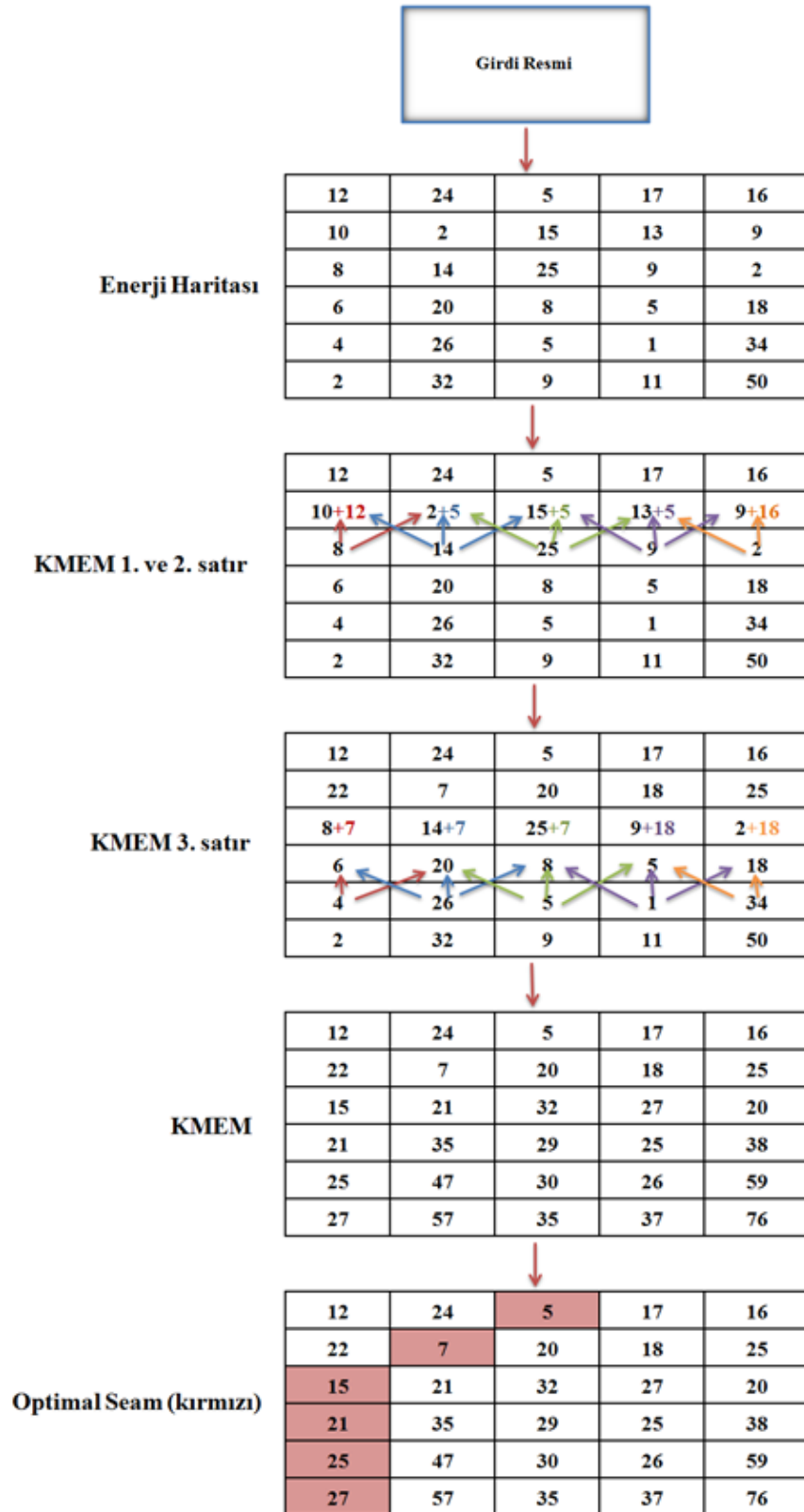
$$s^d = \{x(i), i\}_{i=1}^n \mid \forall i, |x(i) - x(i-1)| \leq 1 \quad (2.2)$$

$$s^y = \{y(j), j\}_{j=1}^m \mid \forall j, |y(j) - y(j-1)| \leq 1 \quad (2.3)$$

Burada  $x: [1, \dots, n] \rightarrow [1, \dots, m]$  ve  $y: [1, \dots, m] \rightarrow [1, \dots, n]$  şeklindedir. En düşük enerjili piksel yolu ise enerji haritasından oluşturulan Kümülatif Minimum Enerji Matrisi (KMEM) yardımıyla elde edilmektedir. KMEM, enerji haritasının ikinci satırından başlanarak en alta kadar gezinilmesi ile elde edilmektedir. Enerji haritasının ilk satırı KMEM'ye aynen geçirilir. Sonraki satırlar ise Denklem 2.4'de verildiği gibi üst minimum komşuların değerlerinin ilgili pikselle toplanması ile oluşturulur.

$$KMEM(i, j) = f_{en}(i, j) + \min(KMEM(i-1, j-1), KMEM(i-1, j), KMEM(i-1, j+1)) \quad (2.4)$$

Dikey bir piksel yolu için KMEM'nin en alttaki satırının minimum elemanına bakılır ve onun minimum komşuları üzerinden ilk satıra kadar gelinir. Böylece minimum enerjiye sahip piksel yolu (optimal piksel yolu) bulunur. Şekil 2.3.'de piksel yolu bulma işlemi görsel olarak açıklanmıştır. Yatay piksel yolu de benzer şekilde bulunmaktadır.



Şekil 2.3. Optimal piksel yolunun bulunması

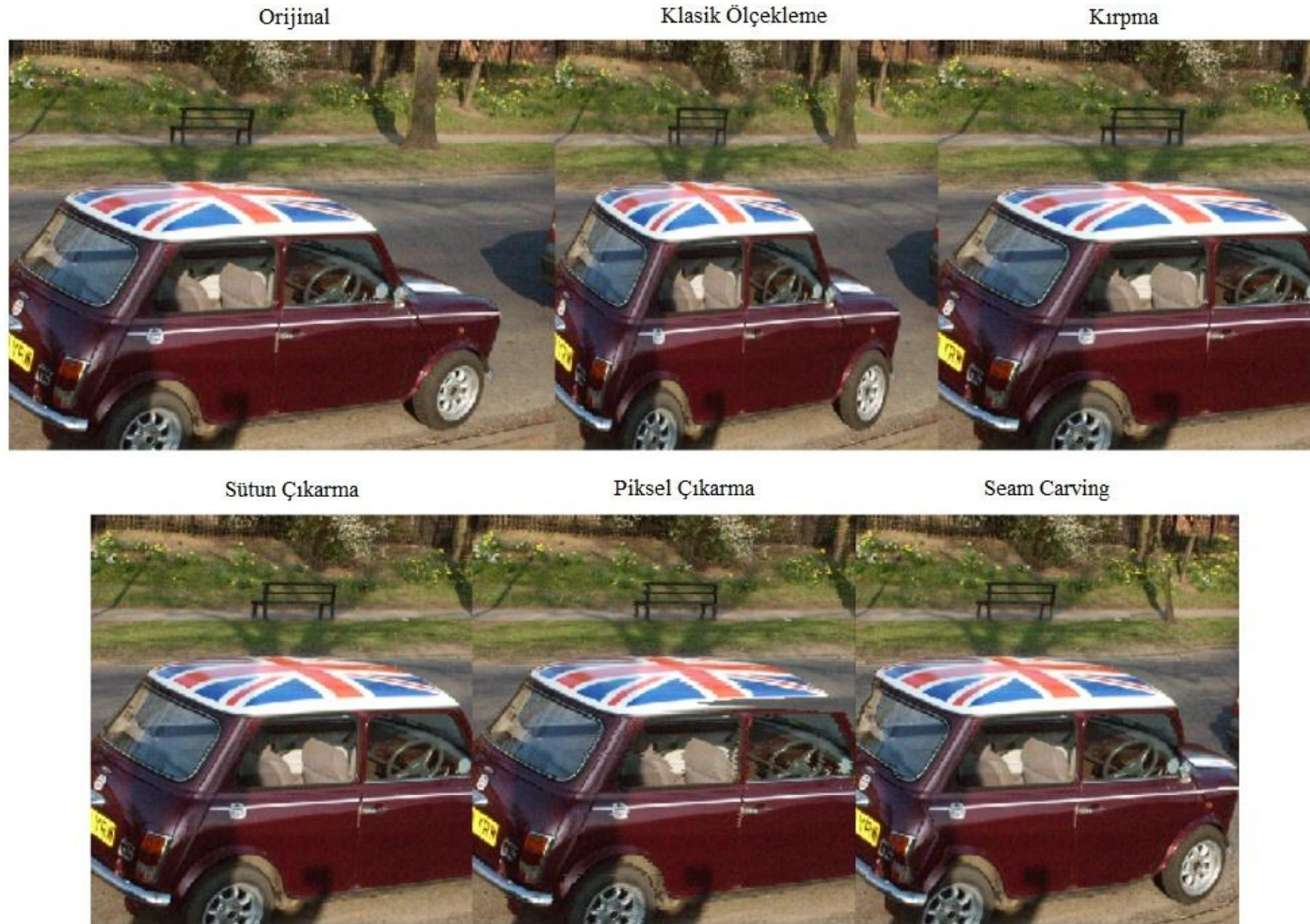


Şekil 2.4. Görüntünün genişliğinin azaltılmasına yönelik beş farklı yaklaşım: klasik ölçekleme, kırpma, sütun çıkarma, piksel çıkarma ve piksel yolu çıkarma



Şekil 2.4. (Devamı)





Şekil 2.4. (Devamı)

İçerik duyarlı ölçeklemedeki farklı yöntemlerin etkinlik açısından değerlendirilmesinde piksellerin ortalama enerjileri incelenebilir. Pikselleri rastgele çıkarırken ortalama enerji değişmemelidir; fakat içerik duyarlı ölçeklemede düşük enerjili pikseller çıkarılıp yüksek enerjili olanlar bırakıldığı için ortalama enerji artmalıdır. Her satırdan belli sayıda pikseli çıkarmak optimal sonucu vermektedir. Fakat bu yöntemle yapılan ölçeklemede görsel uyum bozulmaktadır. Kırpma, enerji korunumu ilkesini en çok bozan yöntem olarak görülmektedir. Sütunların çıkarılması enerji korunumu bakımından daha iyidir; fakat görsel bozukluklara sebep olmaktadır. Piksel yolu çıkarma, görsel bütünlük ve enerji korumu dengesini en iyi sağlayan yöntem olarak görünmektedir. Şekil 2.5.'de farklı görüntüler üzerinde piksel yolu çıkarma ve klasik ölçeklemenin etkileri gösterilmiştir. Şekilde görüldüğü gibi piksel yolu çıkarma görüntünün içeriğini bozmadan ölçeklemeyi gerçekleştirmiştir. Çıkarılan piksel yolları insan yüzü, araba, saat gibi önemli ön plan nesnelere üzerinden değil, nispeten önemsiz arka plan nesnelere üzerinden geçmektedir. Böylece, klasik ölçeklemenin aksine önemli ön plan nesnelere ölçeklenen görüntülerde hala önemini korurken, piksel kayıpları insan gözü tarafından algılanamayacak arka plan nesnelere üzerinden yaşanır.

Piksel yolu çıkarma literatürde farklı amaçlarla da kullanılmıştır. Görüntünün içerik duyarlı ölçeklenmesine [7], [45], [46], [16], [47], [48] ek olarak görüntülerin iyileştirilmesinde [2] ve nesne yok etmede [1] de kullanılmıştır.

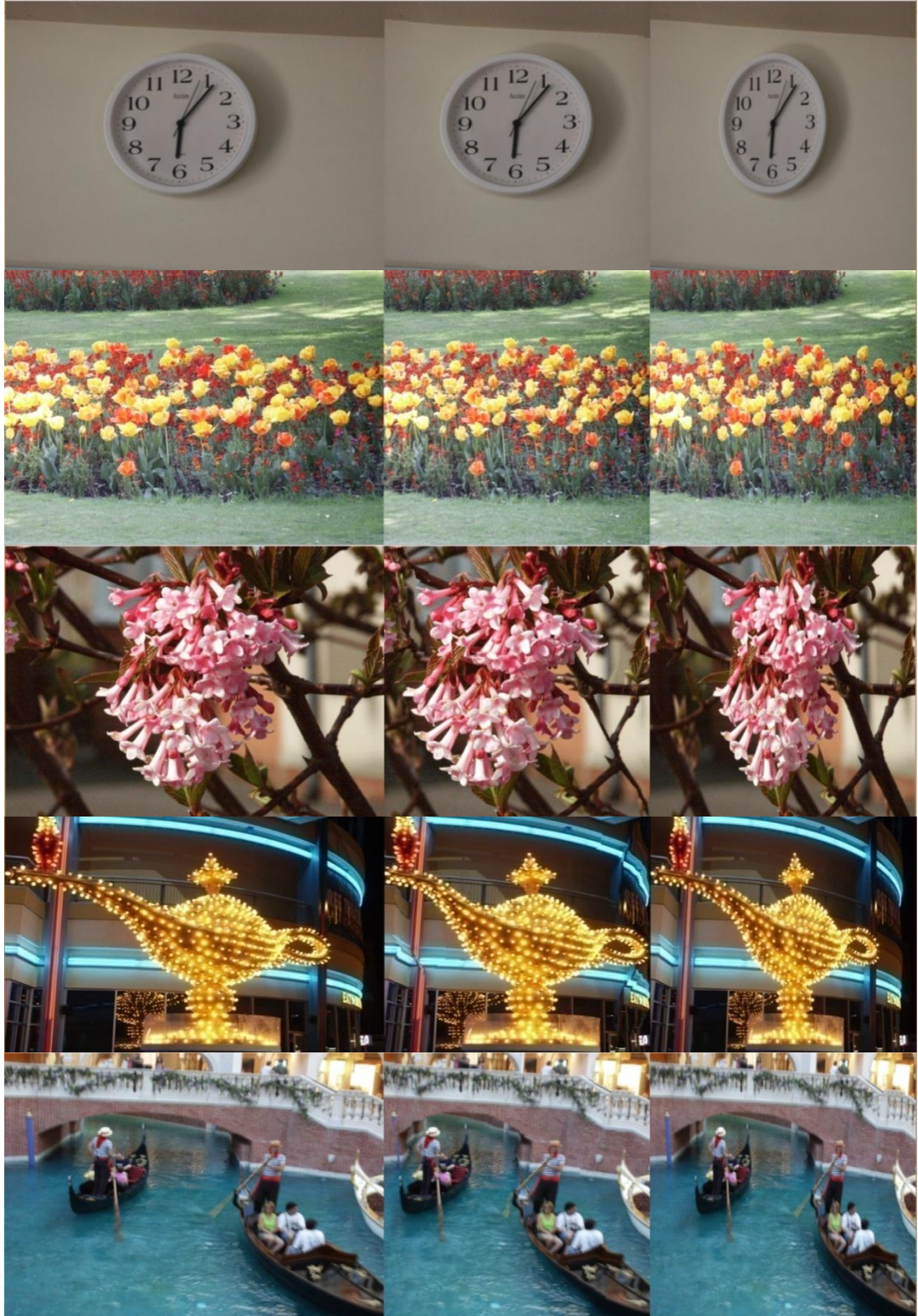


Şekil 2.5. Orijinal görüntü, piksel yolu çıkarma yöntemi ile %30 ölçeklenmiş görüntü ve klasik (eş oranlı) ölçekleme ile %30 ölçeklenmiş görüntü

Orijinal görüntü

Piksel yolu çıkarma

Eş oranlı ölçekleme



Şekil 2.5. (Devamı)



Şekil 2.5. (Devamı)

Orijinal görüntü

Piksel yolu çıkarma

Eş oranlı ölçekleme

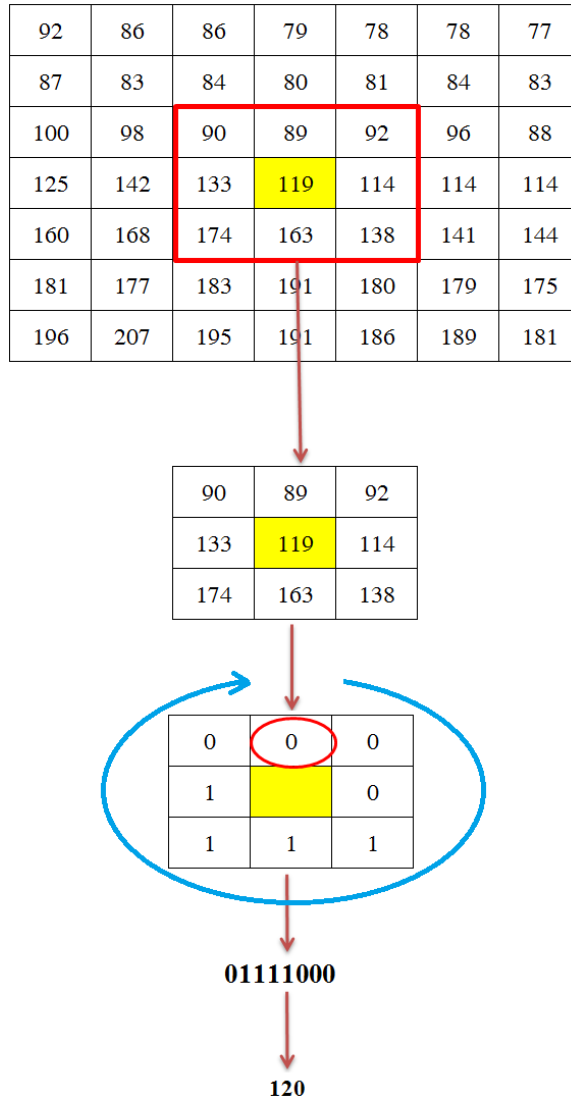


Şekil 2.5. (Devamı)

## 2.2. Yerel İkili Örüntüler (YİÖ)

Yerel İkili Örüntüler (Local Binary Patterns) 1994'te sınıflandırma amaçlı kullanılmak üzere önerilen bir görsel tanımlayıcıdır [49]. Görüntüler genellikle  $3 \times 3$ 'lük hücelere bölünür ve hücre kenarlarındaki her piksel merkezdeki pikselle karşılaştırılır. Bu karşılaştırma sonucunda 8 bitlik bir ikili sayı elde edilir. Eğer pikselin değeri merkez pikselin değerinden büyükse 1, küçükse 0 yazılır. Bu 1 ve 0'lar saat yönünde yan yana yazılarak ikili sayı oluşturulur (Şekil 2.6.). Merkez piksel tam merkezde olmak kaydıyla  $3 \times 3$ 'den farklı pencereler de kullanılabilir. Yani, tek koşul tüm komşuların merkez piksele eşit mesafede yer almasıdır. Şekil 2.7.'de farklı pencere ölçüleri ile YİÖ örnekleri görülmektedir. Bu tez çalışması kapsamında yapılan deneylerde  $3 \times 3$  pencere kullanılmıştır. Şekil 2.6.'de piksel yoğunluk değerleri verilen örnek görüntünün 119 değerine sahip hücresinin YİÖ karşılığının hesaplanması gösterilmiştir.

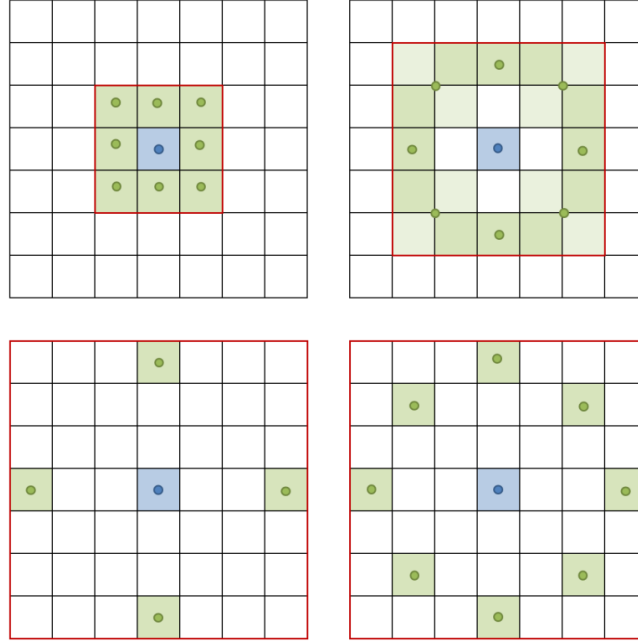
YİÖ literatürde üç boyutlu MRI görüntülerinden beyin tümörlerinin tespiti [50], mamografi görüntülerinden kötü huylu kitlelerin tespiti [51], yüz resimlerinden cinsiyet tespiti [52], [53], yüz tanıma [54], doku sınıflandırma [53], [55], [56] sürücülerin yüz ifadelerinden yorgunluklarının tespiti [57], göz tanıma [58], kelebek türlerinin otomatik sınıflandırılması [59], virüs görüntülerinin sınıflandırılması [60], kişilerin yaşlarının tahmin edilmesi [61], yüz güzelliğine göre sınıflandırma [62] vb. farklı alanlarda kullanılmış ve başarılı sonuçlar elde edilmiştir.



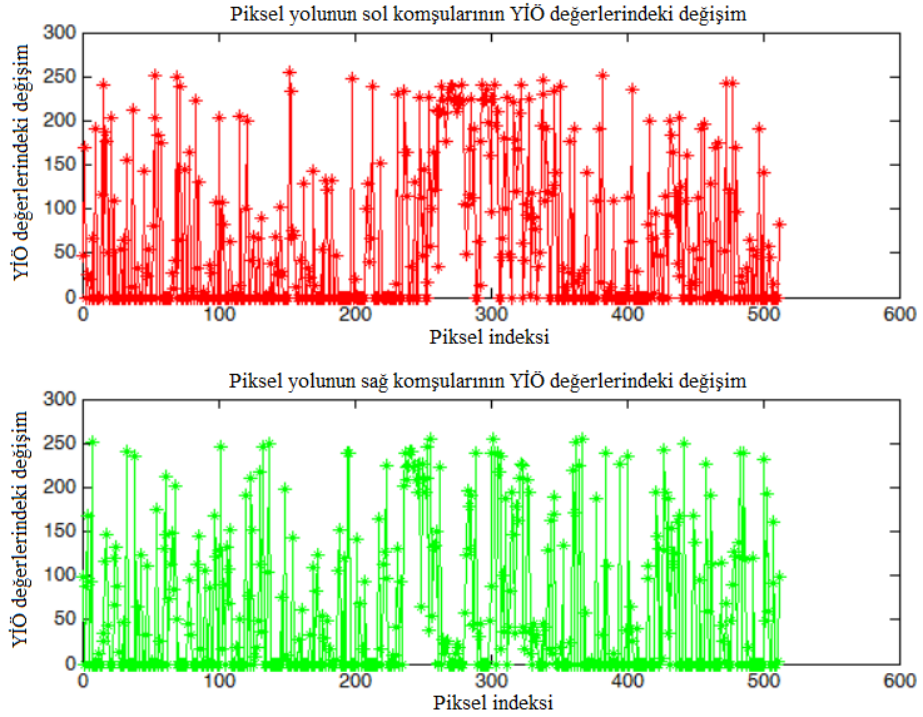
Şekil 2.6. Örnek piksel için YİÖ dönüşümü

Piksel yolu çıkarma tabanlı görüntü boyutlandırma YİÖ kullanmadaki mantık şudur: piksel yolu çıkarma, görüntülerden bir piksel genişliğinde bağlantılı piksel yollarının hedef boyuta erişene kadar birer birer çıkarılması şeklinde uygulanır. Çıkarılan piksel yolunun sağındaki pikseller sola (altındaki pikseller üste) bir kaydırılır. Dolayısıyla, oluşacak bozukluklar çıkarılan piksel yoluna (seam'e) komşu piksellerde oluşacaktır. Yani, resmin genel dokusunda değil yerel bölgelerinde bozukluklar olacaktır. Bu durumda yerel bir doku tanımlayıcı kullanmak, oluşacak bu bozuklukları belirgin hale getirecek, dolayısıyla da piksel yolu çıkarmayı ele verecektir [39]. Piksel yolu çıkarma ile genişlik azaltma işleminden sonra çıkarılan piksel yoluna komşu piksellerin YİÖ değerlerindeki değişim Şekil 2.8.'de

gösterilmiştir. Piksel yolunun sol ve sağ komşularının piksel yolu çıkarma öncesi ve sonrası değerleri arasındaki fark YİÖ tabanlı piksel yolu çıkarma tespit yönteminin kayda değer sonuç üreteceğinin bir göstergesidir.



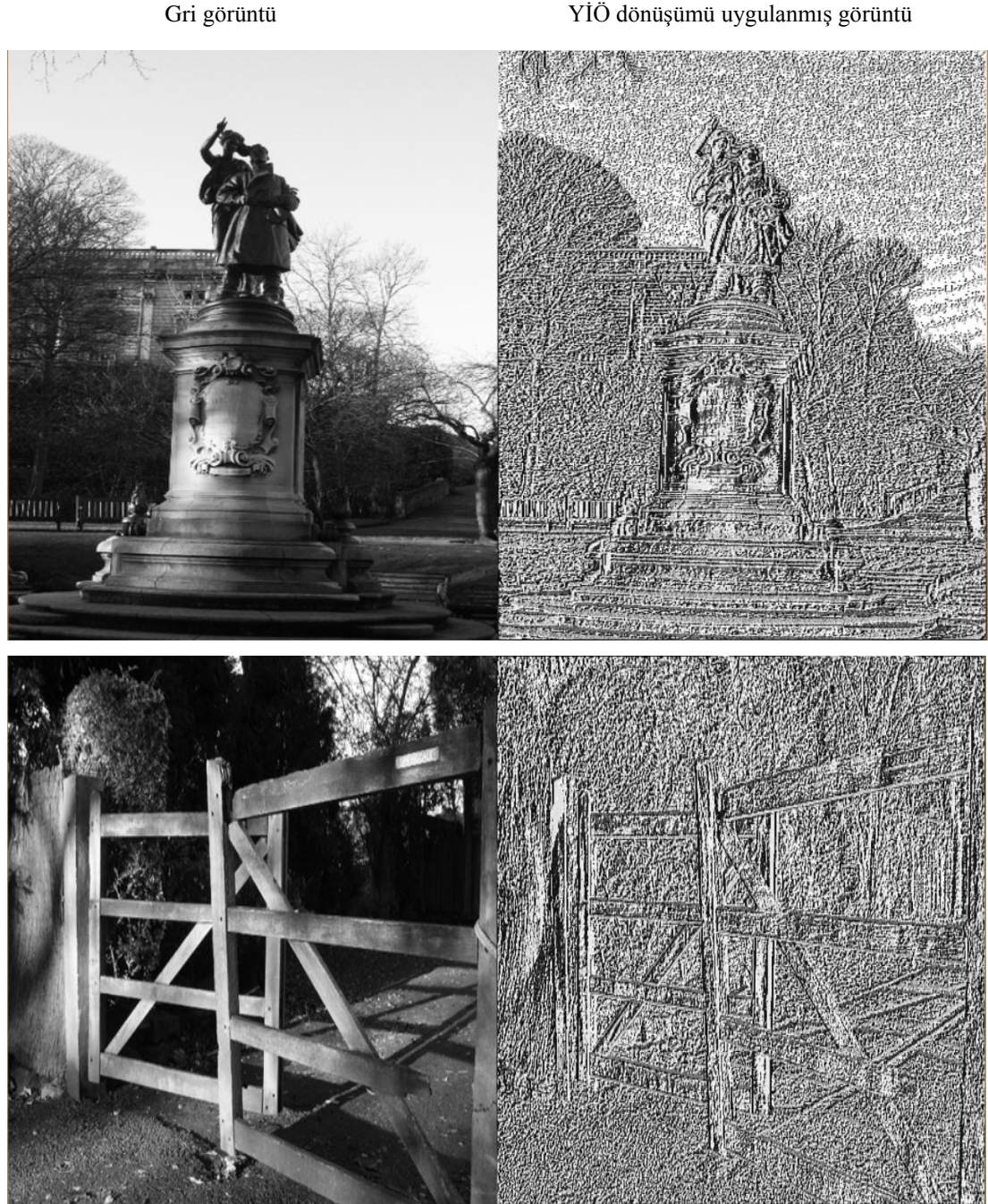
Şekil 2.7. Farklı komşuluk mesafesine ve komşu sayısına sahip YİÖ örnekleri



Şekil 2.8. Piksel yolu çıkarma öncesi ve sonrası piksel yoluna komşu piksellerin YİÖ değerlerindeki değişim



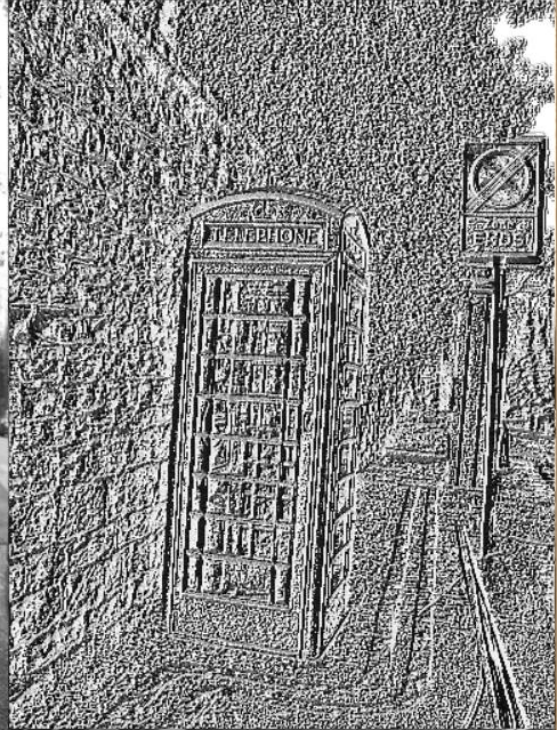
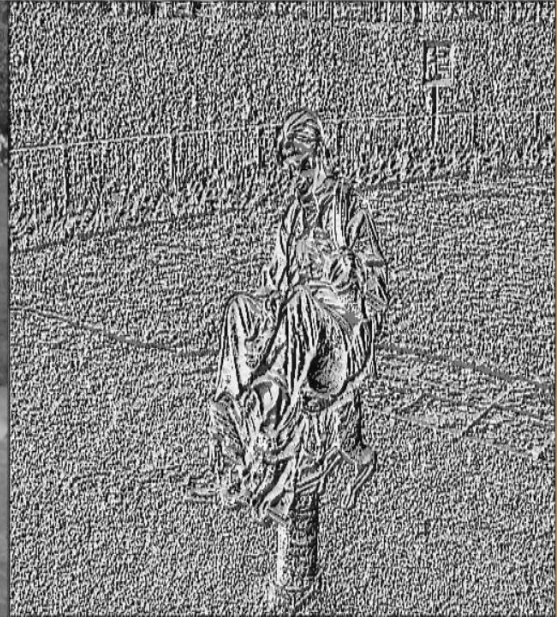
Önerilen tespit yaklaşımının uygulamalarında kullanılmak üzere veri kümesindeki tüm görüntüler öncelikle YİÖ dönüşümüne tabi tutulur, ardından uygulanacak özellik çıkarımı işlemleri gerçekleştirilir. Veri kümesinden örnek görüntüler ve bu görüntülerin YİÖ dönüşümleri Şekil 2.9.'da gösterilmiştir.



Şekil 2.9. Gri görüntü (solda) ve YİÖ dönüşümü (sağda)

Gri görüntü

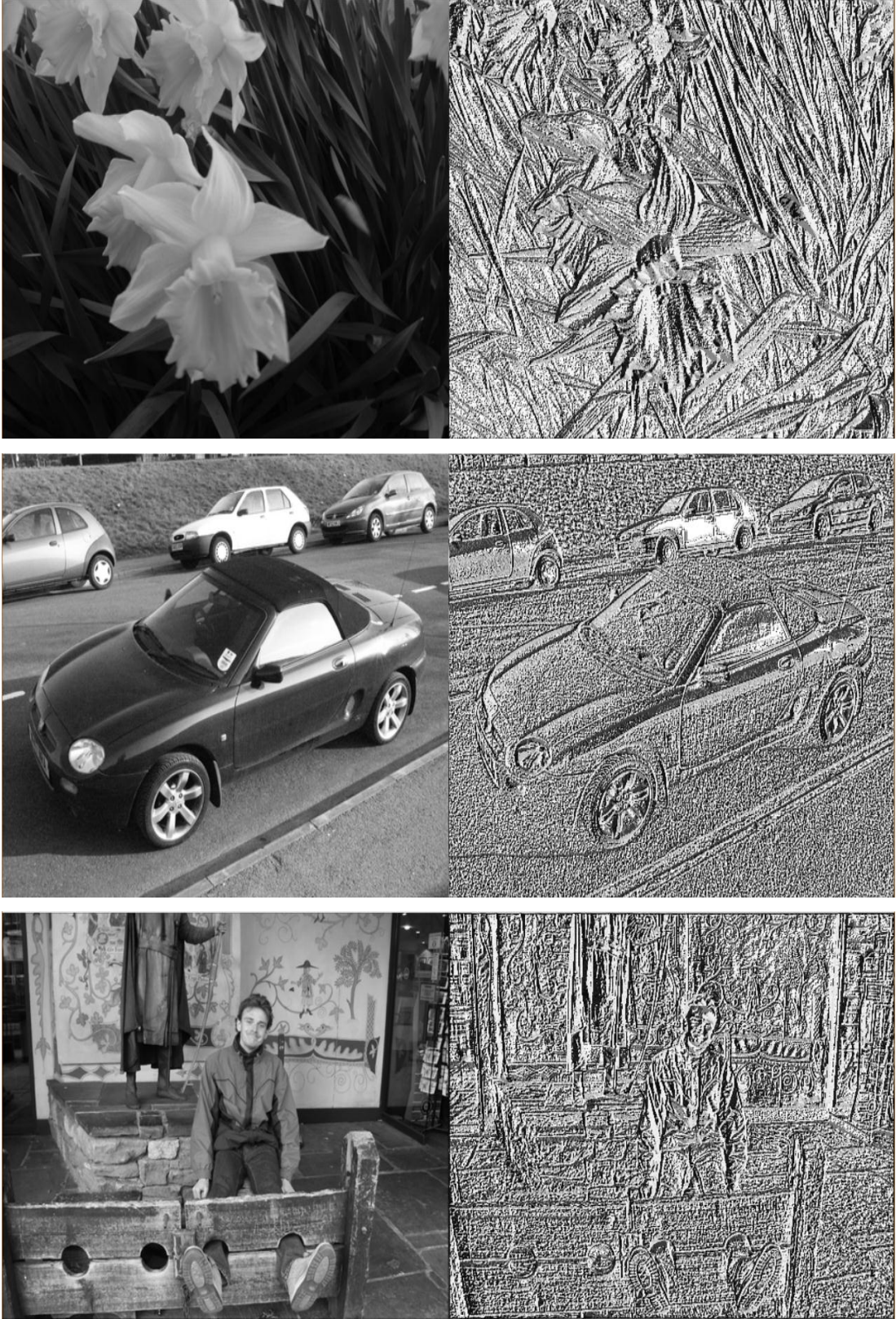
YİÖ dönüşümü uygulanmış görüntü



Şekil 2.9. (Devamı)

Gri görüntü

YİÖ dönüşümü uygulanmış görüntü

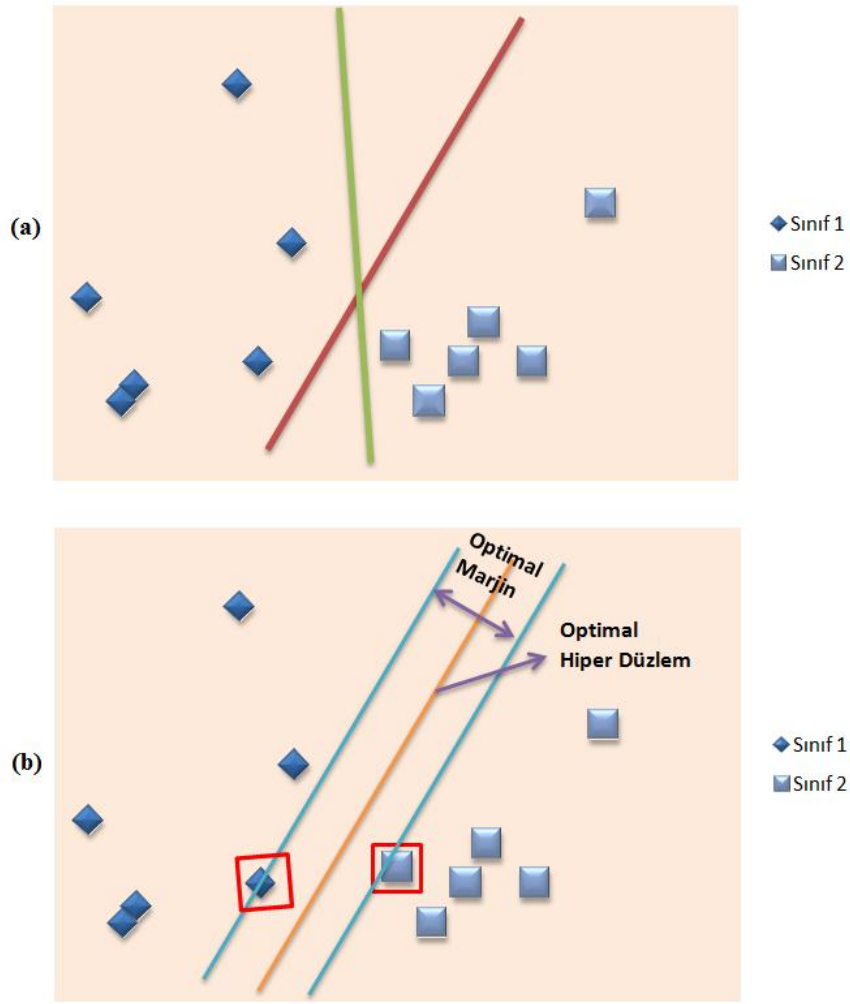


Şekil 2.9. (Devamı)

### 2.3. Destek Vektör Makineleri (DVM)

Destek Vektör Makineleri bir dizi etiketli eğitim verisinden giriş çıkış haritalama fonksiyonları üreten bir danışmanlı öğrenme yöntemidir [63]. Haritalama fonksiyonu bir sınıflandırma fonksiyonu (giriş verisinin kategorisi) ya da bir regresyon fonksiyonu olabilir. DVM, farklı sınıflardaki birbirine en yakın noktalar arasındaki mesafeyi (marjini) maksimize eden bir hiper düzlem tanımlamaktadır. Bu tanımlama lineer olmayan ayrılabilir (nonlinearly seperable) sınıflandırma problemlerine genişletilebilir. Veri, lineer karar düzlemi ile sınıflandırılabilir şekilde yüksek boyutlu bir uzaya haritalanır. Bu noktada lineer olmayan çekirdek fonksiyonları görev alır ve orijinaline göre daha ayrılabilir bir veri seti elde edilir. Daha sonra, bahsedilen hiper düzlem oluşturulur. İdeal bir DVM analizi öyle bir hiper düzlem oluşturur ki bu düzlem, vektörleri (durumları) sınıflar birbiriyle hiç çakışmayacak şekilde ayırır. Fakat mükemmel ayırım her zaman mümkün olmayabilir ya da oluşan modelin sınıflandırmayı doğru yapmadığı çok fazla durum olabilir. Böyle durumlarda DVM marjini maksimize eden hiper düzlemi bulur ve yanlış sınıflandırmaları en aza indirir.

DVM istatistiksel öğrenme teorisindeki güçlü matematiksel temelini yanında, biyoinformatik, metin madenciliği, yüz tanıma ve görüntü işleme gibi pek çok uygulamada oldukça yüksek performans sergilemektedir [63]. DVM'nin başarısı sınıfları birbirinden ayıracak optimal düzlemi belirleyen noktaların seçiminden gelmektedir. Lineer regresyon ve acemi Bayes yöntemlerinde tüm noktalar optimalliği belirlerken, DVM'de sadece "zor noktalar" , yani karar sınırına yakın olan noktalar optimalliği belirler [64]. Şekil 2.8.'de baklava dilimi ve kare şekillerde iki sınıfa ait veri ve muhtemel iki ayırıcı düzlem (a) ve DVM'nin bulduğu hiper düzlem (b) görülmektedir. Bulunan optimal düzlem iki sınıfın birbirine en yakın üyelerine (zor noktalar) eşit mesafededir, optimal marjini ortalar.



Şekil 2.10. (a) İki sınıflı veri kümesi ve muhtemel ayırıcı düzlemler (b) DVM'nin "zor noktalar"a (kırmızı çerçeveli) göre belirlediği optimal hiper düzlem

DVM'nin amacı, eğitim verisine bağlı olarak, test verisinin öznelikleri verilmişken bunlara ait hedef değerlerini (sınıflarının) tahmin eden bir model üretmektir. Yöntemde üç farklı durumla karşılaşılabılır:

1. Lineer bir sınıflayıcı ve lineer ayrılabilen problem
2. Lineer bir sınıflayıcı ve tamamen lineer ayrılamayan problem
3. Lineer olmayan sınıflayıcı ve lineer ayrılamayan problem

Aşağıdaki bölümlerde bu problemler ve çözümleri detaylı olarak incelenecektir.

### 2.3.1. Lineer bir sınıflayıcı ve lineer ayrılabilen problem

Cortes ve Vapnik [65] tarafından geliştirilen DVM ile çözülebilecek en basit problem türü lineer ayrılabilen bir örnek uzayını sınıflandırmak için lineer bir sınıflayıcı kullanmaktır. Böyle bir örnek uzayında, sınıfları ayıracak en iyi hiper düzlemi bulmak istediğimizi varsayalım. Eğitim kümesinde  $L$  tane nokta, her  $x_i$  girdisinin  $D$  tane özniteliği ve ait oldukları  $y_i = -1$  ya da  $y_i = +1$  şeklinde iki sınıf olsun. Bu durumda eğitim verisi Denklem 2.5'deki gibi olur.

$$\{x_i, y_i\} \mid i = 1, \dots, L \text{ ve } y_i \in \{-1, 1\}, x \in \mathcal{R}^D \quad (2.5)$$

Veri lineer ayrılabilir olduğundan  $D = 2$  için  $x_1$ 'e karşılık  $x_2$  grafiği üzerine her iki sınıfı birbirinden ayıran bir doğru,  $D > 2$  olduğu durumda ise  $x_1, x_2, x_3, \dots, x_D$ 'lerin grafiği üzerine iki sınıfı ayıran bir hiper düzlem çizilebilir (Şekil 2.11.).

Bu hiper düzlem  $w \cdot x + b = 0$  şeklinde tanımlanabilir. Burada  $w$  hiper düzlemin normali,  $\frac{b}{|w|}$  ise düzlemden orijine olan dik uzaklığı ifade etmektedir. Destek vektörleri, ayırıcı hiper düzleme en yakın olan örneklerdir ve DVM'nin amacı bu hiper düzlemi, her iki sınıfın birbirine en yakın üyelerinden mümkün olduğunca uzağa yerleştirmektir. Bu durumda;

$$w \cdot x_i + b \geq +1 \quad \forall x_i \in \text{Sınıf 1}, \text{yani } y_i = +1 \quad (2.6)$$

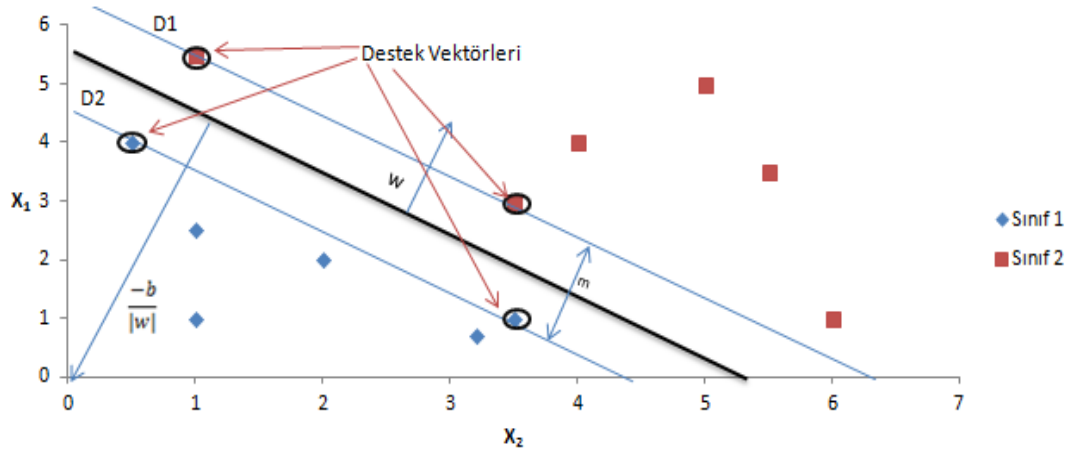
$$w \cdot x_i + b \leq -1 \quad \forall x_i \in \text{Sınıf 2}, \text{yani } y_i = -1 \quad (2.7)$$

eşitsizlikleri sağlanır. Denklemlerdeki nokta, nokta çarpımını ifade eder ve  $w \cdot x_i = w^T x_i$  anlamına gelir. Bu denklemler Denklem 2.8'deki gibi birleştirilebilir.

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad \forall i \quad (2.8)$$

Destek vektörlerinden geçen  $D_1$  ve  $D_2$  düzlemlerinin denklemleri sırasıyla Denklem

2.9 ve 2.10'daki gibi olacaktır.



Şekil 2.11. Lineer ayrılabilen iki sınıfı ayıran hiper düzlem [66]

$$w \cdot x_i + b = +1 \quad (2.9)$$

$$w \cdot x_i + b = -1 \quad (2.10)$$

Şekil 2.11.'de destek vektörlerinden geçen bu paralel düzlemler arasındaki mesafe  $m$  ile gösterilmiştir.  $m$ , DVM'nin marjini olarak ifade edilen,  $D_1$  ve  $D_2$  düzlemleri arasındaki mesafeyi ortalar. Ayrıcı hiper düzlemi DVM'nin amacına uygun şekilde yerleştirebilmek için bu marjinin maksimize edilmesi gerekmektedir. Basit vektör geometrisinden marjinin  $\frac{1}{\|w\|}$  olduğu çıkarılabilir.  $\frac{1}{\|w\|}$ 'nin maksimize edilmesi de  $\|w\|$ 'nin minimize edilmesi demektir. Minimizasyonda Denklem 2.8'deki koşul sağlanmalıdır. Kuadratik programlama optimizasyonuna imkân sağlayabilmek için minimizasyon problemi şöyle düşünülebilir:

$$\min \frac{1}{2} \|w\|^2 : y_i(w \cdot x_i + b) - 1 \geq 0 \quad \forall i \quad (2.11)$$

Bu minimizasyon problemindeki kısıtları sağlayabilmek için Lagrange çarpanları ( $\alpha$ )  $\alpha_i \geq 0 \quad \forall i$  olmak üzere eklenir:

$$\mathcal{L} \equiv \frac{1}{2} \|w\|^2 - \alpha [y_i(w \cdot x_i + b) - 1 \quad \forall i] \quad (2.12)$$

$$\equiv \frac{1}{2} \|w\|^2 - \sum_{i=1}^L \alpha_i [y_i(w \cdot x_i + b) - 1] \quad (2.13)$$

$$\equiv \frac{1}{2} \|w\|^2 - \sum_{i=1}^L \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^L \alpha_i \quad (2.14)$$

Burada Denklem 2.14'ü minimize edecek  $w$  ve  $b$ 'ye ve maksimize edecek  $\alpha$ 'ya ( $\alpha_i \geq 0 \quad \forall i$ ) ihtiyaç vardır. Bunu sağlamak için de  $\mathcal{L}$ 'nin  $w$  ve  $b$ 'ye göre kısmi türevleri alınarak sıfıra eşitlenir.

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^L \alpha_i y_i x_i \quad (2.15)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^L \alpha_i y_i = 0 \quad (2.16)$$

Denklem 2.15 ve 2.16, Denklem 2.14'de yerine yazılır.

$$\mathcal{L} \equiv \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^L \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad : \quad \alpha_i \geq 0 \quad \forall i, \quad \sum_{i=1}^L \alpha_i y_i = 0 \quad (2.17)$$

Denklem 2.17, yalnız  $\alpha_i$ 'ye bağlı bir fonksiyondur ve bu probleme dual problem denir (eğer  $w$  bilirse tüm  $\alpha_i$ 'ler bilinebilir, eğer tüm  $\alpha_i$ 'ler bilirse  $w$  bilinebilir). Dual problemlerin hedef fonksiyonu maksimize edilmelidir. Dolayısıyla, problem Denklem 2.18'de verilen şekle dönüşmektedir.

$$\text{maks. } W(\alpha) = \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^L \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad : \quad \alpha_i \geq 0 \quad \forall i, \quad \sum_{i=1}^L \alpha_i y_i = 0 \quad (2.18)$$

Denklem 2.16 bir kuadratik programlama problemidir ve  $\alpha$  kolaylıkla bulunabilir.  $\alpha$  bilindiğinde de Denklem 2.15'den  $w$  elde edilir. Buraya kadar hiper düzlemi elde



edebilmek için bir tek  $b$ 'nin hesaplanması kalıyor. Bu noktada da destek vektörlerini kullanabiliriz. Destek vektörleri  $D_1$  ve  $D_2$  düzlemleri üzerinde yer alır (siyahla işaretlenmiş örnekler) ve Denklem 2.12'deki eşitliği sağlar.

$$y_d(w \cdot x_d + b) = 1 \quad (2.19)$$

Denklem 2.19'da  $x_d$  ve  $y_d$  destek vektörlerinin değer ve sınıfını ifade etmektedir. Denklem 2.15 burada yerine yazılırsa,

$$y_d \left( \sum_{m \in D} \alpha_m y_m x_m \cdot x_d + b \right) = 1 \quad (2.20)$$

elde edilir. Burada  $D$  destek vektörlerinin  $\alpha_d > 0$ 'ı sağlayan indislerini içeren kümedir. Denklem 2.18'in her iki tarafı  $y_d$  ile çarpılırsa  $y_d^2 = 1$  olacağından,

$$b = y_d - \sum_{m \in D} \alpha_m y_m x_m \cdot x_d \quad (2.21)$$

elde edilir. Rastgele bir destek vektörü kullanmak yerine  $b$ 'yi hesaplarırken destek vektörlerinin ortalamasını kullanmak daha iyi olur. Dolayısıyla, en genel hali ile  $b$  sabiti,

$$b = \frac{1}{N_d} \sum_{d \in D} \left( y_d - \sum_{m \in D} \alpha_m y_m x_m \cdot x_d \right) \quad (2.22)$$

şeklinde elde edilir.  $w$  ve  $b$  değişkenleri elde edildiğine göre ayırıcı hiper düzlemin optimal yerleşimi, yani Destek Vektör Makinesi (DVM) oluşturulabilir. Her yeni nokta  $x'$ ,

$$y' = \text{sign}(w \cdot x' + b) \quad (2.23)$$

işaret fonksiyonu ile değerlendirilerek sınıflandırılabilir.

### 2.3.2. Lineer bir sınıflayıcı ve tamamen lineer ayrılamayan problem

Bir önceki başlıkta anlatılan DVM yöntemini, Şekil 2.12.'deki gibi tamamen lineer ayrılmayan veri setlerine genişletmek için Denklem 2.6 ve 2.7'de verilen kısıtları genişletmek gerekmektedir ki yanlış sınıflandırılacak örnekleri kendi sınıflarına dâhil edebilsin. Bu da yeni bir pozitif esneklik değişkeninin eklenmesi ile yapılır:

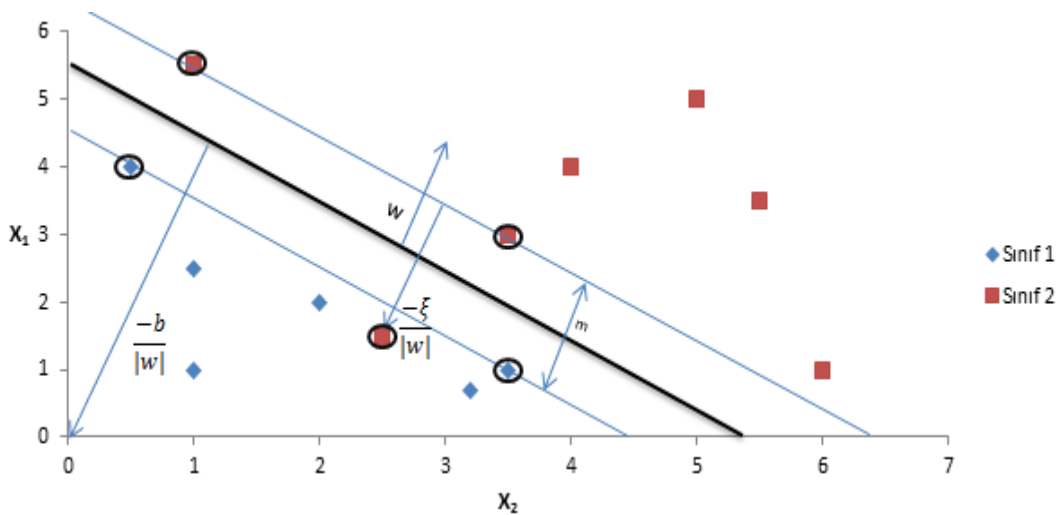
$$w \cdot x_i + b \geq +1 - \xi_i, \quad y_i = +1 \quad (2.24)$$

$$w \cdot x_i + b \leq -1 + \xi_i, \quad y_i = -1 \quad (2.25)$$

$$\xi_i \geq 0 \quad \forall i \quad (2.26)$$

Bu üç denklem şu şekilde birleştirilebilir:

$$y_i(w \cdot x_i + b) - 1 + \xi_i \geq 0 \quad : \quad \xi_i \geq 0 \quad \forall i \quad (2.27)$$



Şekil 2.12. Tamamen lineer ayrılmayan veri kümesi ve lineer ayıran hiper düzlem [66]

Bu problem türünde marjin sınırlarının yanlış tarafında olan örnekler marjine olan uzaklığına bağlı olarak artan cezalarla cezalandırılırlar. Yanlış sınıflandırılan örnek sayısını azaltmak amacıyla Denklem 2.11'deki hedef fonksiyonu Denklem 2.28'deki gibi güncellenir.

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^L \xi_i \quad : \quad y_i(w \cdot x_i + b) - 1 + \xi_i \geq 0 \quad \forall i \quad (2.28)$$

Denklemdaki C parametresi esneklik değişkeninin ceza değeri ile marjin uzunluğu arasındaki dengeyi kontrol eder. Eldeki minimizasyon problemini çözmek üzere önceki bölümde olduğu gibi  $w$ ,  $b$  ve  $\xi_i$ 'ye göre minimize edilecek,  $\alpha$ 'ya göre  $\alpha_i \geq 0, \mu_i \geq 0 \forall i$  koşulunu sağlayarak maksimize edilecek Lagrange formülü oluşturulur:

$$\mathcal{L} \equiv \frac{1}{2} \|w\|^2 + C \sum_{i=1}^L \xi_i - \sum_{i=1}^L \alpha_i [y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^L \mu_i \xi_i \quad (2.29)$$

$w$ ,  $b$  ve  $\xi_i$ 'ye göre türev alınarak sıfıra eşitlendiğinde:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^L \alpha_i y_i x_i \quad (2.30)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^L \alpha_i y_i = 0 \quad (2.31)$$

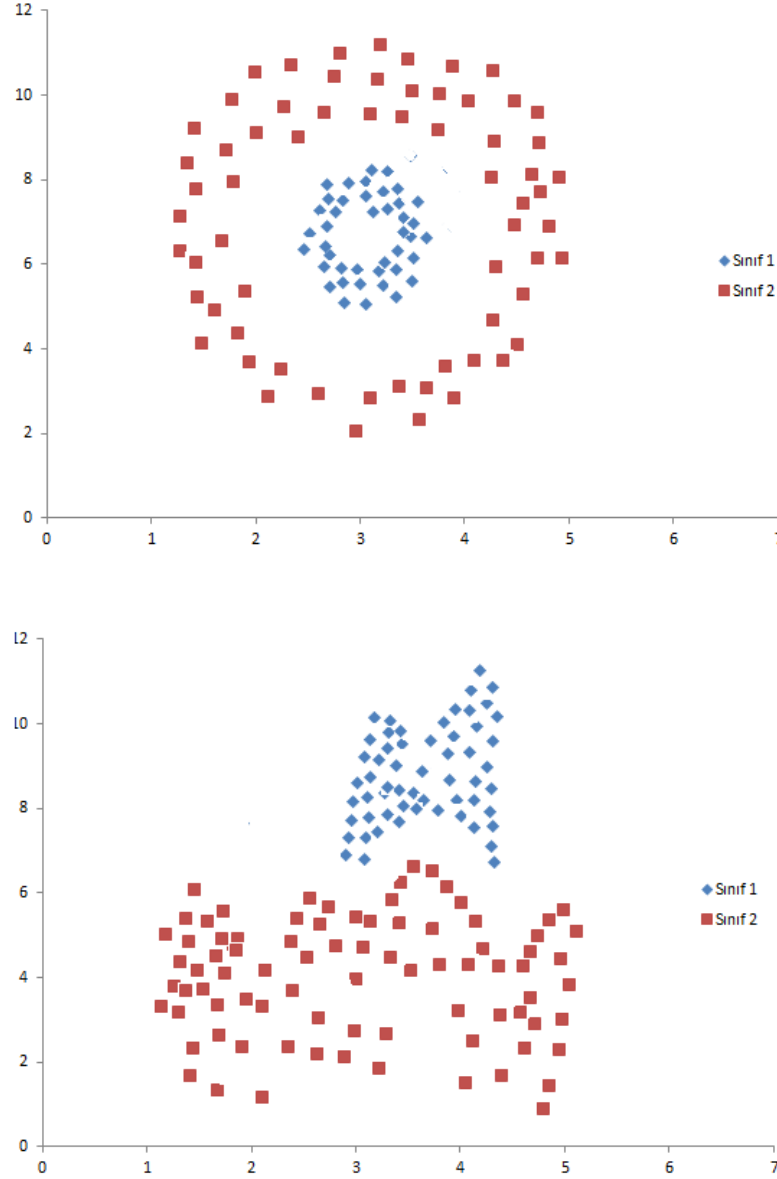
$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \mu_i \quad (2.32)$$

denklemleri elde edilir. Bu sonuçlar yerine yazıldığında Denklem 2.17 ile aynı denklem elde edilir; fakat Denklem 2.32'nin etkisiyle kısıt değişir.  $C = \alpha_i + \mu_i$  ve  $\mu_i \geq 0$  olduğundan  $\alpha \leq C$  olur. Dolayısıyla, yeni maksimizasyon problemi,

$$\begin{aligned} \text{maks. } W(\alpha) &= \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^L \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad : \quad 0 \leq \alpha_i \\ &\leq C \quad \forall i, \quad \sum_{i=1}^L \alpha_i y_i = 0 \end{aligned} \quad (2.33)$$

şekline gelir. Daha sonra  $b$  önceki problemle aynı şekilde hesaplanır. Hesaplama  $0 \leq \alpha_i \leq C$  'yi sağlayan indisler kullanılır.

### 2.3.3. Lineer olmayan sınıflayıcı ve lineer ayrılmayan problem



Şekil 2.13. Radyal Temelli Çekirdek ile yeniden haritalanarak ikiye ayrılmış veri [66]

Şimdiye kadar anlatılan lineer ayrılabilen DVM'yi uygularken kullanılan nokta çarpımları

$$k(x_i, x_j) = x_i \cdot x_j = x_i^T x_j \quad (2.34)$$

şeklindedir. Her nokta çarpımı aslında Denklem 2.34'de verilen çekirdek fonksiyonuna karşılık gelir. Bu fonksiyon Lineer Çekirdek olarak bilinir. Çekirdek fonksiyonları lineer ayırlamayan veri setini lineer olmayan bir ayırıcı düzlemle ayrılabilir şekilde daha yüksek bir özellik uzayına haritalayan fonksiyonlardır. Farklı çekirdek fonksiyonları vardır. Bu fonksiyonlar Denklem 2.35'den türetilmiştir ve iki vektörün iç çarpımına dayalıdır. Bu da demek oluyor ki, eğer fonksiyonlar lineer olmayan özellik haritalama fonksiyonu  $x \mapsto \phi(x)$  ile daha yüksek boyutlu bir uzaya dönüştürülebiliyorsa, özellik uzayındaki haritalanan girdilerin yalnızca iç çarpımlarının hesaplanması gerekir, açıkça  $\phi$ 'in hesaplanmasına gerek yoktur [66].

$$k(x_i, x_j) = e^{-\left(\frac{\|x_i - x_j\|^2}{2\alpha^2}\right)} \quad (2.35)$$

Şekil 2.13.'de solda iki boyutlu veri uzayında lineer ayırlamayan bir veri, Denklem 2.35'deki çekirdek fonksiyonu ile kapalı olarak tanımlanan, lineer olmayan özellik uzayında (sağda) ayrılabilir. Bu fonksiyon Radyal Temelli Çekirdek olarak bilinir. Sınıflandırmada kullanılan diğer popüler çekirdek fonksiyonları Polinom ve Sigmoid çekirdekleridir ve sırasıyla Denklem 2.36 ve Denklem 2.37'de verilmişlerdir.

$$k(x_i, x_j) = (x_i \cdot x_j + a)^b \quad (2.36)$$

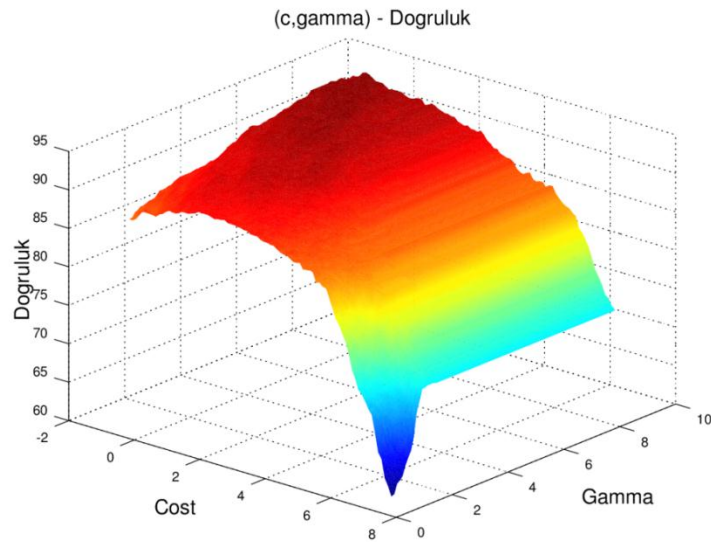
$$k(x_i, x_j) = \tanh(ax_i \cdot x_j - b) \quad (2.37)$$

Denklemlerdeki a ve b çekirdeğin davranışını tanımlayan parametrelerdir.

#### 2.4. Izgara Arama (Grid Search) Yöntemi

Hiper parametreler, bir tahmin edici (sınıflayıcı) içinde doğrudan öğrenilemeyen parametrelerdir. Genellikle, sınıflayıcının yapıcı fonksiyonuna parametre olarak geçirilir. Bu parametrelerin tipik örneği ise Destek Vektör Makinelerinin *Cost* (c),

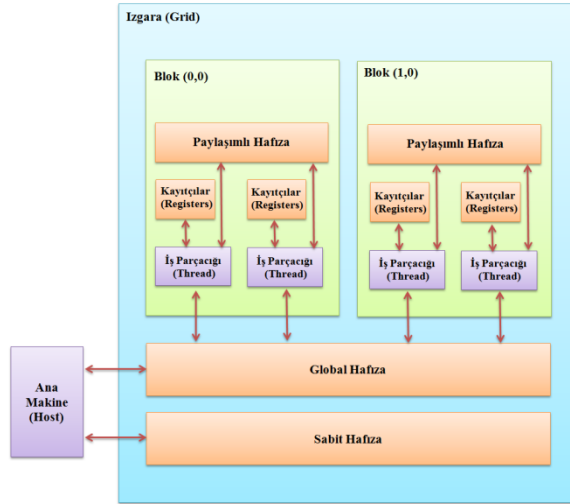
*çekirdek* ve *Gamma* parametreleridir. Izgara arama yöntemi parametrelerin muhtemel tüm kombinasyonlarını değerlendirmeye katar ve her bir parametre kombinasyonu için sınıflayıcının performansı test edilir. Farklı *c* ve *gamma* ( $\gamma$ ) değerleri için doğruluktaki değişim Şekil 2.14.'de gösterilmiştir. Parametrelerin veri aralığı çok geniş denendiği durumlarda parametrelerin uzunlukları çarpımı kadar kombinasyon olacağından optimal parametrelerin bulunması çok uzun sürebilmektedir. Özellikle de eğitim ve test kümelerindeki örnek sayısı fazlaysa sınıflayıcının mevcut parametrelere göre sınıflandırma yapması saatler alabilmektedir. Bu tez çalışmasında ölçeklenmiş ve orijinal çok sayıda veriye ait 210 özellik çıkarılmış ve örneğin karışık veri kümesi ile yöntemin test edilmesinde  $10.030 \times 210$  boyutlarında eğitim verisi kullanılmıştır. DVM'nin *c* ve *gamma* ( $\gamma$ ) parametreleri için en az 20'şer değer ile yöntem test edilerek optimal parametre seçilmeye çalışılmıştır. Dolayısıyla,  $20 \times 20 \times 20 \times 20$  muhtemel ( $c, \gamma$ ) ikilisi için  $10.030 \times 210$  boyutlarında veri ile test yapılmıştır. Bu miktarda veri ve deneme sayısı için CPU (Central Processing Unit- Merkezi İşlem Birimi) kullanarak işlemleri seri olarak yapmak oldukça uzun işlem zamanı gerektirmektedir. Bu kaybı gidermek ve önerilen yöntemi daha etkin test edebilmek için ızgara arama yöntemi GPU (Graphics Processing Unit- Grafik İşlem Birimi) ile hızlandırılmıştır.



Şekil 2.14. ( $c, \gamma$ ) değiştiğinde doğruluktaki değişim

### 2.4.1. GPU ile hızlandırılmış ızgara arama

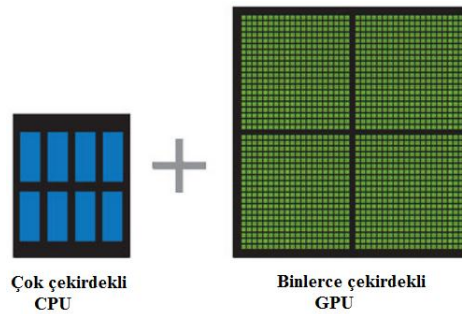
Geleneksel olarak, GPU'lar bilgisayarlar ve oyun konsolları için adanmış olarak kullanılmaktaydı. Fakat günümüzde eğilim daha çok programlama ve grafik öncelikli uygulama geliştirme yönünde değişmiştir [67]. Hesaplama yoğunluklu ve çok sayıda paralel işlemin olduğu durumlarda GPU özelleşmiş renderleme (rendering) donanımı sayesinde CPU'ya göre daha avantajlıdır. CPU'larda ardışık seri işlemler için optimize edilmiş az sayıda çekirdek varken, bir GPU'nun çoklu görevleri eş zamanlı olarak gerçekleştirebilecek daha küçük ve daha etkili binlerce çekirdeği vardır (Şekil 2.16.). Modern GPU'lardaki fazla sayıdaki aritmetik birim, pek çok işlem yoğunluklu uygulamayı hızlandırmada kullanılabilir [68]. GPU tabanlı uygulama geliştirmede farklı üreticilerin CUDA (Compute Unified Device Architecture-NVIDIA) ve CTM (Close To Metal-AMD) platformları mevcuttur[69]. Bu platformlarda grafik kartının donanın fonksiyonlarına doğrudan erişim hesaba katılmıştır. CUDA, C programlama dilinin bir açılımı iken; CTM, assembler kodlarını derleyen bir sanal makinedir. Bu çalışmada NVIDIA'nın CUDA platformu kullanılmıştır. CUDA programcıya GPU'nun donanım fonksiyonlarını daha iyi kontrol edebilme imkânı sunar [69]. CUDA hafıza türleri ve birimler arasındaki iletişimin yönlerini içeren mimari yapısı Şekil 2.15.'de verilmiştir. GPU için gerçekleştirilen kodlamalar çekirdek denen C fonksiyonlarının oluşturulması ile gerçekleştirilir [67]. Donanım üzerinde bir zamanda yalnızca bir çekirdek çalıştırılabilir ve yapılandırılan tüm iş parçacıkları (threads) çekirdeği paralel olarak çalıştırır. İş parçacıkları ızgara içindeki bloklarda gruplandırılmıştır. Bir iş parçacığı başlatıldığında, ızgaranın blokları uygun derleme kapasitelerine göre çoklu işlemcilerle dağıtılır. Bloktaki iş parçacıkları tek bir çoklu işlemci üzerinde eş zamanlı olarak derleme yapar. Her bir iş parçacığı bloğu işini bitirdiği zaman çoklu işlemciler serbest bırakılır ve onun yerine yeni bir blok başlatılır [67].



Şekil 2.15. CUDA mimarisi [70]

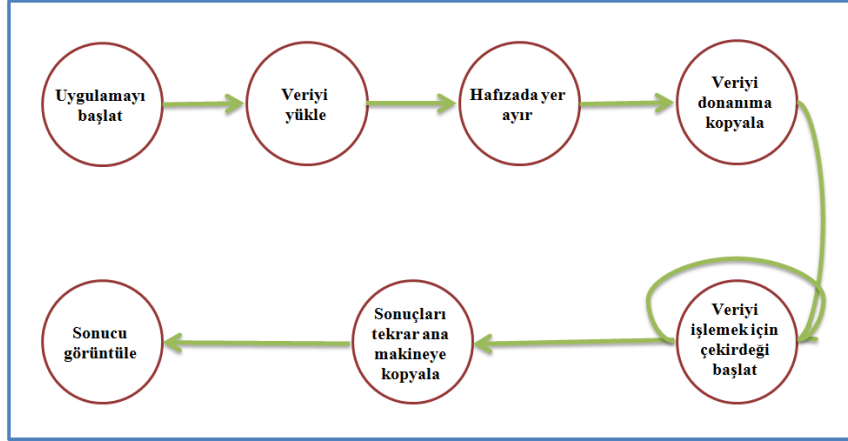
CUDA ayrıca aynı bloktaki iş parçacıkları arasında kısıtlı senkronizasyon sağlar. Bu senkronizasyon, `syncthreads()` fonksiyonunun çağırılması ile gerçekleştirilir. Fonksiyon çağırıldığında her iş parçacığı, çağrı tüm iş parçacıklarına iletilene kadar beklerler. Böylelikle, blok içindeki iş parçacıkları arasındaki iletişim koordine edilip, paylaşım ve global hafızada olabilecek veri okuma yazma riskleri minimize edilir [67].

Pek çok CUDA uygulaması bir dizi program akışını takip eder. Öncelikle ana makine veriyi bir kaynaktan (örneğin bir metin dosyasından) yükler ve bir veri yapısı içinde ana makinenin hafızasında depolar. Daha sonra veri için bir donanım hafızasını tahsis eder ve veriyi buraya kopyalar. Sonra, veriyi işlemek ve sonuç üretmek üzere çekirdekler başlatılır. Daha sonra bu sonuçlar görüntülemek ya da daha ileri işlemler için tekrar ana makineye kopyalanır (Şekil 2.17.) [67].



Şekil 2.16. CPU-GPU[71]



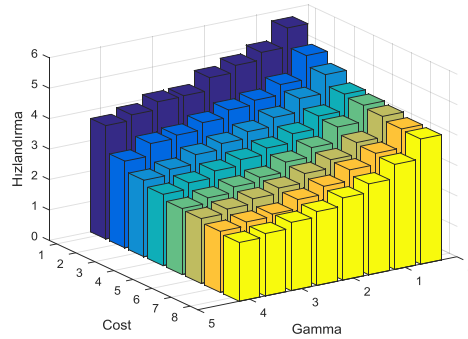


Şekil 2.17. CUDA program akışı [67]

Önerilen piksel yolu çıkarma esaslı boyut değişikliği tespit yaklaşımında karşılaşılan en büyük problem gerçekleştirilen işlemlerin çok zaman almasıdır. Önceki başlıkta anlatıldığı gibi, fazla sayıda veri ile fazla sayıda çevrimi gerçekleştirmek büyük zaman kayıplarına sebep olmaktadır. Yaşanan bu zaman kayıplarını giderebilmek için DVM'nin optimal parametrelerine karar verme aşamasında CPU yanında geniş aralıklarda parametre aramalarında GPU ile hızlandırma da kullanılmıştır. Yapılan uygulamalarda Austin Carpenter tarafından yazılan cuSVM kütüphanesi [72] kullanılmıştır. cuSVM, MEX dosyaları aracılığıyla CUDA (Compute Unified Device Architecture) fonksiyonlarının çağrılması şeklinde çalışır. Kütüphanede eğitim ve test işlemlerini gerçekleştirecek sırasıyla cuSVMTrain ve cuSVMPredict fonksiyonları bulunmaktadır. Eğitim fonksiyonu girdi olarak sınıf etiketleri, eğitim verileri, DVM'nin  $c$ ,  $\gamma$  ve *çekirdek* parametrelerini alır. Sonuçta da destek vektörlerinin katsayılarını, DVM tahmin fonksiyonundaki  $b$  ofsetini ve destek vektörlerinin matrisini döndürür. Tahmin fonksiyonu ise test verilerine ait matrisi, cuSVMTrain'den dönen destek vektörleri matrisi ve destek vektörlerinin katsayıları, ofset ve çekirdek parametresini girdi olarak alır. cuSVMPredict fonksiyonu çıktı olarak ise verilen giriş vektörüne karşılık cuSVMTrain'de oluşturulan modeli kullanarak elde edilen tahmin vektörünü döndürmektedir.

Şekil 2.18.'de piksel yolu çıkarma tabanlı görüntü ölçeklemenin tespiti için karışık

veri kümesi ile test deneylerinin GPU kullanımına kullanımı ile hızlandırmaya ait bir örnek verilmiştir. Burada farklı Cost (c) ve Gamma ( $\gamma$ ) parametreleri için yaklaşık 2-6 kat arası hızlandırma elde edilmiştir. Gerçekleştirilen deneylerde kullanılan bilgisayar Nvidia GeForce GTX 960M işlemciye sahip bir kişisel bilgisayardır.



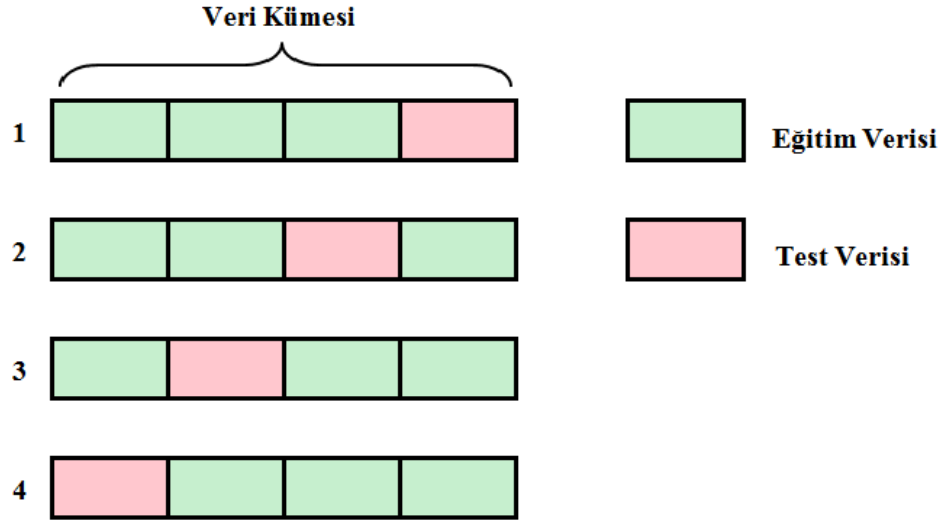
Şekil 2.18. Farklı Cost ve Gamma değerleri için GPU ile elde edilen hızlandırma

## 2.5. K Katlı Çapraz Sağlama (K-Fold Cross Validation)

Çapraz sağlama, öğrenme algoritmalarını değerlendirmek ve karşılaştırmak için eldeki veriyi eğitim ve testte kullanılmak üzere ikiye bölerek bir kısmı ile (eğitim) modelin eğitilmesi, kalan kısmı ile de modelin sağlamlasının yapılması şeklinde gerçekleştirilen istatistiksel bir yöntemdir [73]. Bir model seçildiğinde, bu modelin performansının belirlenmesi için eldeki verilerin gerçeğe en yakın olacak şekilde, belli bir dağılım olmaksızın rastgele belirlenmesi gerekir. Ancak bu şekilde performansı değerlendirilen model gerçek başarısını gösterebilir.  $k$  katlı çapraz sağlamada veri kümesi  $k$  alt kümeye bölünür ve model  $k$  kere test edilir. Her seferinde  $k$  alt kümeden biri test için  $k-1$ 'i de eğitim için kullanılır. Daha sonra  $k$  denemenin ortalama hatası hesaplanır. Veri kümesindeki her nokta yalnızca bir kere test setinde olurken  $k-1$  kere eğitim setinde bulunur.  $k$  arttıkça tahmin değerinin varyansı azalır. Yöntemin dezavantajı, modelin  $k$  kere çalıştırılıyor olmasıdır. Bu, işlem süresini  $k$  kat artırır.

Bu tez çalışmasında, önerilen piksel yolu çıkarma esaslı tespit yöntemi için eğitim ve test kümelerinin belirlenmesinde dört katlı çapraz sağlama yöntemi kullanılmıştır. Bu

amaçla veri seti dört eşit parçaya bölünmüş ve parçaların üçü eğitimde, biri testte kullanılmıştır (Şekil 2.19.). Muhtemel dört kombinasyon için model çalıştırılmış ve ortalama doğruluk tablolarına yansıtılmıştır. Modelin her testinde, eğitimde kullanılmayan veriler kullanılmıştır.



Şekil 2.19. Dört Katlı Çapraz Sağlama

## 2.6. ROC (Receiver Operating Characteristics) Analizi

ROC grafiği, sınıflayıcıları performanslarına göre görselleştirmek, düzenlemek ve seçmek için kullanılan bir yöntemdir [74]. ROC grafikleri uzun yıllar sinyal algılama teorisinde sınıflayıcıların yanlış alarm oranı ile isabet oranı arasındaki ilişkiyi göstermek için kullanılmıştır [75], [76]. ROC analizi daha sonra teşhis sistemlerinin analizinde kullanılmaya başlamıştır [77]. Medikal karar verme durumlarında çok fazla sayıda çalışmada ROC eğrileri kullanılmış ve halen de kullanılmaktadır [78], [79], [80], [81], [82]. ROC eğrilerinin makine öğrenmesinde kullanımı ise 1980'lerin sonlarında başlamaktadır [83] ve günümüzde de pek çok makine öğrenmesi ile alakalı çalışmada analizler bu grafikler ile gerçekleştirilmektedir [84], [85], [86], [87].

İki sınıflı bir sınıflandırma problemi ele alındığında, her bir örnek pozitif ve negatif sınıf etiketlerinden birine ( $\{p,n\}$  etiket kümesi) haritalanır. Sınıflandırma modeli, örneklerden tahmin edilen sınıflara yapılan bir haritalamadır. Bazı modeller bir sınıfa

aitlik olasılığı gibi sürekli çıktı üretirler ki böyle durumlarda belli bir eşik değeri kullanılarak sınıf üyeliği tahmin edilebilir. Diğer modeller ise ayrık sınıf etiketleri üretirler ve bu etiketler eldeki örneğin hangi sınıfa ait olduğunun tahminidir.

Bir sınıflayıcı ve bir örnek veri verildiğinde dört muhtemel sonuç elde edilir. Eğer örnek pozitif ise ve pozitif olarak sınıflandırılırsa *doğru pozitif* olarak sayılır. Negatif olarak sınıflandırılmışsa *yanlış negatif* olarak sayılır. Eğer örnek negatif ise ve negatif olarak sınıflandırılmışsa *doğru negatif*, pozitif olarak sınıflandırılmışsa *yanlış pozitif* olarak sayılır. Bir sınıflayıcı ve bir grup örnek verildiğinde Şekil 2.20.'deki 2×2 karmaşıklık matrisinin (confusion matrix) oluşturulması pek çok yaygın performans ölçütünün temelini oluşturur. Bazı ölçütler Denklem 2.38 - Denklem 2.44'de verilmiştir.

$$YPO = \frac{YP}{YP + DN} \quad (2.38)$$

$$DPO = \frac{DP}{DP + YN} \quad (2.39)$$

$$YNO = \frac{YN}{DP + YN} \quad (2.40)$$

$$DNO = \frac{DN}{YP + DN} \quad (2.41)$$

$$Hassasiyet = \frac{DP}{DP + YN} \quad (2.42)$$

$$Doğruluk = \frac{DP + DN}{DP + YN + YP + DN} \quad (2.43)$$

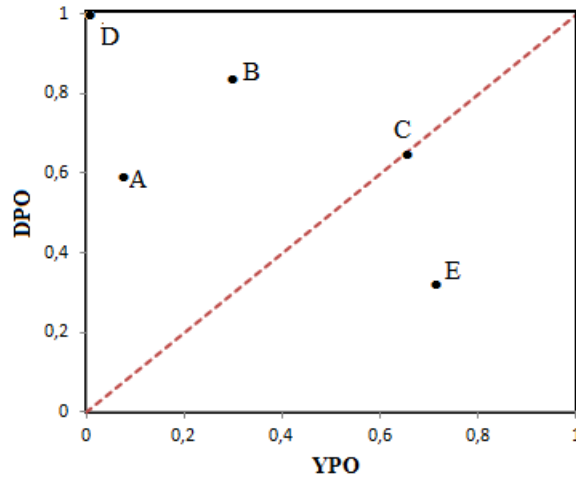
$$Özgüllük = \frac{DN}{DN + YP} \quad (2.44)$$

Denklemlerde kısaltmalar YP: Yanlış Pozitif sayısı, YN: Yanlış Negatif sayısı, DP: Doğru Pozitif sayısı, DN: Doğru Negatif sayısı, YPO: Yanlış Pozitif Oranı, YNO: Yanlış Negatif Oranı, DPO: Doğru Pozitif Oranı, DNO: Doğru Negatif Oranı şeklindedir.

		Gerçek Sınıflar	
		Pozitif	Negatif
Tahmin edilen Sınıflar	Pozitif	<b>Doğru Pozitif</b>	<b>Yanlış Pozitif</b>
	Negatif	<b>Yanlış Negatif</b>	<b>Doğru Negatif</b>
<b>Sütun Toplamı:</b>		<b>Pozitif örnek sayısı</b>	<b>Negatif örnek sayısı</b>

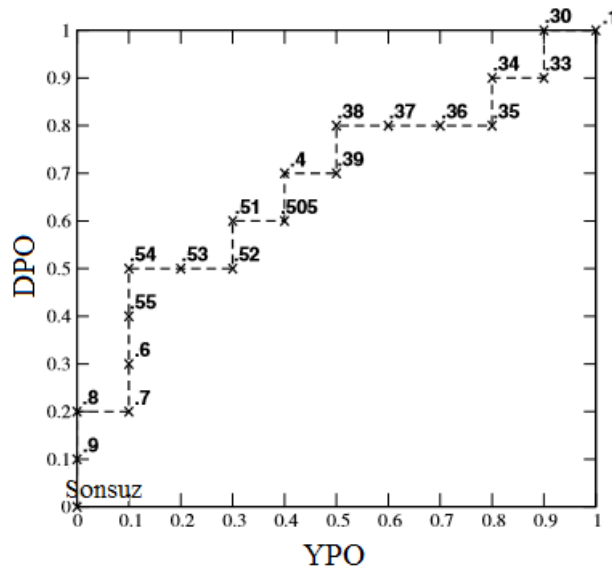
Şekil 2.20. Karmaşıklık matrisi

ROC grafikleri X ekseninde YPO, Y ekseninde DPO'nun olduğu iki boyutlu grafiklerdir. Bir ROC eğrisi fayda maliyet dengesini gösterir. Şekil 2.21.'de beş ayrıklı sınıflandırıcıyı gösteren bir ROC eğrisi verilmiştir. Ayrıklı sınıflandırıcılar yalnızca sınıf etiketi üretirler. ROC uzayında bazı noktalar önemlidir. Sol alt nokta (0,0) hiç pozitif bir sınıflandırma yapılmadığını gösterir. Böyle bir sınıflayıcının yanlış pozitif hatası olmaz; fakat doğru pozitif de olmaz. Tam tersi, sağ üst nokta (1,1) tamamen pozitif sınıflandırma yapar. (0,1) noktası mükemmel sınıflandırmayı gösterir. Şekil 2.21.'deki  $D$ 'nin performansı mükemmeldir. Genel olarak şu söylenebilir: ROC uzayındaki bir nokta eğer kuzey batıya doğru ise diğerlerinden iyidir. Kuzey batıda olmak DPO'nun daha yüksek, YPO'nun daha az olduğunu gösterir. Bu da istenen durumdur. ROC grafiğinin sol tarafındaki X eksenine yakın olan sınıflayıcılar pozitif sınıflandırmayı güçlü bir delil varsa yaparlar. Dolayısıyla, yanlış pozitif hatası çok azdır ama genelde DPO'da düşük olur. Sağ üst taraftaki sınıflayıcılar zayıf bir delil de olsa pozitif sınıflandırma yaparlar. Dolayısıyla, YPO'ları oldukça fazladır. Şekil 2.21.'de A pozitif sınıflandırmayı güçlü bir delil varsa yapmaktadır ve B'ye göre YP'si daha azdır. Pek çok gerçek dünya domeninde negatif örneklerin sayısı fazladır ve sınıflayıcıların ROC grafiğinin uzak sol taraflarında elde edilmesi istenir [74].



Şekil 2.21. Beş ayrıncı sınıflandırıcıyı gösteren ROC eğrisi [74]

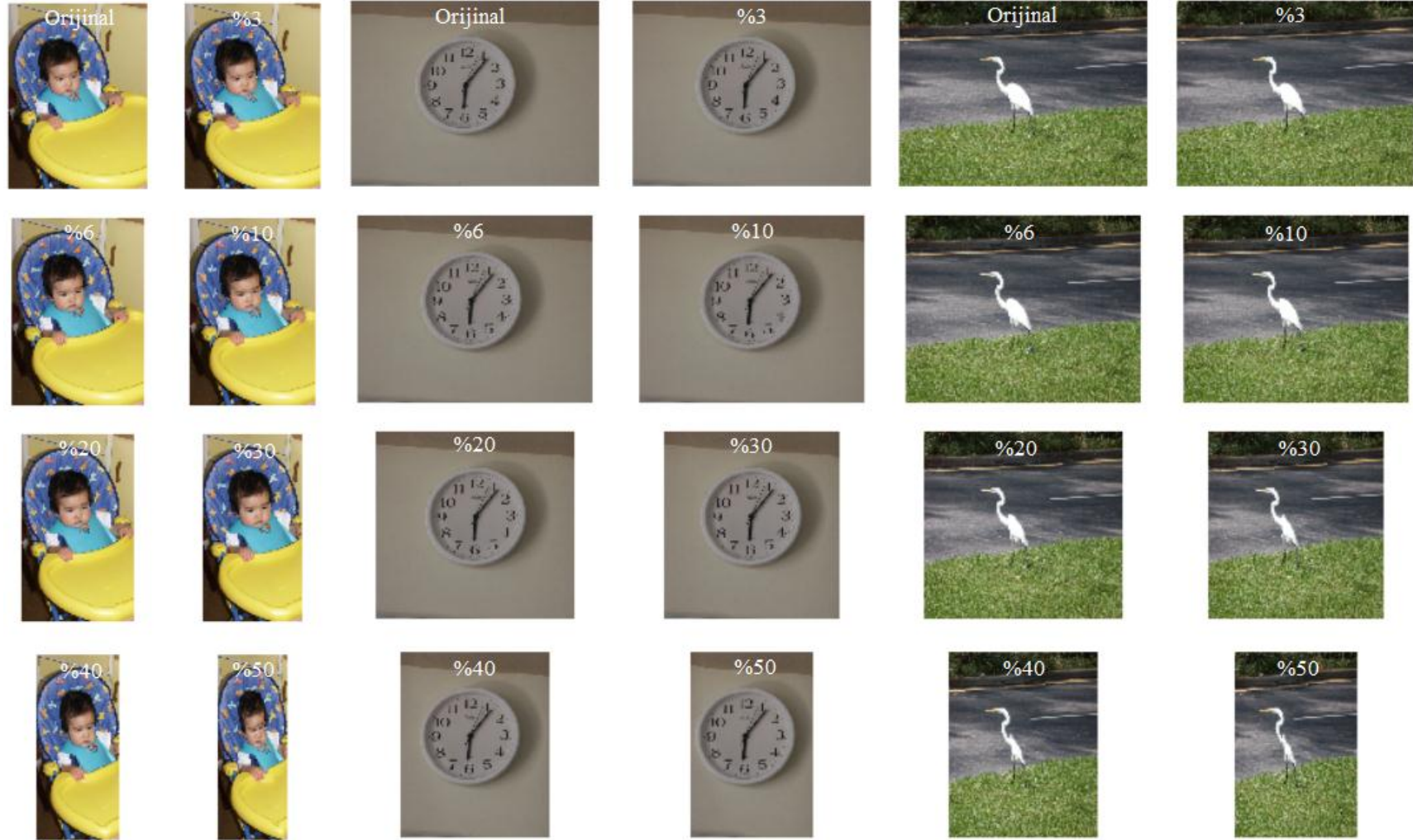
Sonlu bir kümeden üretilen ROC eğrileri aslında bir basamak fonksiyonudur. Kümedeki örnek sayısı sonsuza yaklaştıkça doğru eğriye yaklaşılr. ROC eğrisi, eşikler belirlenerek her eşik değeri için DPO-YPO değerlerinin hesaplanması ile elde edilir. Şekil 2.22.'de örnek ROC eğrisi ve eğri üzerinde eşikler verilmiştir. Eşik değeri sonsuzken hiç pozitif sınıflandırma yapılamamıştır (0,0). Eşik düşürüldükçe eğri yukarı tırmanmış ve 0,1 yapıldığında (1,1) noktasına gelerek sonlanmıştır. Görüldüğü gibi eşik düşürmek DPO'yu artırmıştır.



Şekil 2.22. Örnek ROC eğrisi [74]

## **BÖLÜM 3. HİBRİT VE ŞERİT TABANLI PİKSEL YOLU ÇIKARMA TESPİTİ**

Klasik ölçekleme yöntemlerinin aksine, piksel yolu çıkarma (seam carving) görüntünün boyutlarını değiştirirken yalnızca geometrik özellikleri değil, görüntünün içeriğine bağlı olan anlamsal, piksel önemine bağlı olan özelliklerini de kullanır. Bu sebeple de yapılan değişiklikler çok yüksek miktarda olmadığı sürece insan gözü tarafından algılanamaz. Şekil 3.1.'de örnek bir görüntü ile genişliği piksel yolu çıkarma yöntemi kullanılarak %3, %6, %10, %20, %30, %40 ve %50 azaltılmış görüntüler verilmiştir. Piksel yolu çıkarılarak işlenmiş bu görüntülerde yapılan değişiklik hissedilememektedir. Özellikle de düşük ölçekleme oranlarında (%3 ve %6) işlenmiş görüntü orijinalden ayrılmamaktadır. Dolayısıyla, piksel yolu çıkarmanın özellikle düşük ölçekleme yapıldığı durumlarda tespiti büyük önem arz etmektedir. Bu sebeple, bu tez çalışmasında, piksel yolu çıkarma tabanlı ölçeklemenin tespitinin zor olduğu düşük ölçekleme oranları (%30'a kadar) hesaba katılmıştır. %30'dan daha yüksek ölçekleme oranlarında bozukluklar çok daha belirgin olup insan gözü tarafından algılanabilmektedir ve bunun için derin bir analize gerek yoktur. Tespit için önerilen yaklaşımların başarısını düşük ölçekleme oranlarında elde edilen performans belirlemektedir. Bu çalışmada, piksel yolu çıkarma tabanlı ölçeklemenin tespiti için öncelikle hibrit özelliklerin kullanıldığı bir yöntem önerilmiş, ardından daha yüksek başarımların elde edildiği şerit tabanlı tespit yöntemi geliştirilmiştir.



Şekil 3.1. Piksel yolu çıkarma yöntemi ile ölçeklenmiş görüntüler



### 3.1. Hibrit Pksel Yolu Çıkarma Tespit Yöntemi

Bir görsel tanımlayıcı olarak YİÖ görüntünün dokusundaki yerel değişimleri ortaya çıkarır. Bu sebeple, görüntülerin piksel yoğunluk değerleri yerine YİÖ dönüşümlerini kullanmak tespit oranının artırılmasına katkı sağlar. Tez çalışmasının bu aşamasında özellikle düşük ölçekleme oranlarında daha doğru tespit oranları elde edebilmek için hibrit bir tespit mekanizması önerilmiştir [88]. YİÖ tabanlı ve piksel yoğunlukları tabanlı özellikler çıkarılarak DVM altmış özellik ile eğitilmiştir. Bu yöntemle eski yöntemle göre düşük ölçekleme oranlarında yaklaşık %9 gelişim kat edilmiştir. Yapılan deneyler önerilen hibrit yaklaşımla daha tatmin edici tespit oranlarının elde edilebileceğini göstermiştir.

Bir görüntüden bir ya da daha fazla piksel yolunun çıkarılması, mevcut görüntünün ortalama enerjisini artırır; çünkü düşük enerjili piksel yolları çıkarılır. Dolayısıyla, görüntünün enerjisini incelemek tespit işini kolaylaştırır. Bunun yanında, görüntünün gürültü seviyesi de piksel yolu çıkarılmasından etkilenir; çünkü genellikle düz alanlar görüntüden çıkarılır [32]. Bu sebeplerle, Ryu ve ark.[32]'nin önerdiği on sekiz özelliği kullanmak mantıklıdır. Öte yandan, piksel yolu çıkarma yöntemi görüntüleri mantıksal açıdan ele aldığından işleme sonrası bulanıklaşma, gölge gibi yaygın görüntü boyutlandırma bozuklukları olmaz; fakat çıkarılan ya da eklenen piksel yollarının çevresinde yerel bozukluklar oluşur. Bu bozuklukların belirginleşmesini sağlayacak yerel doku tanımayıcı, YİÖ, kullanmak tespit doğruluğunu artırabilir. Pkseller arası ilişki ve eş dizimlilik matrisleri de bu bozuklukları ortaya çıkarabilse de piksel yolu çıkarma yöntemi için etkili değildirler [39]. Bu düşünce, yukarıda bahsedilen özelliklerin YİÖ tabanında kullanılmasına yönelmektedir. Yin ve ark. görüntünün üst yarısının optimal piksel yolu ile tamamı baz alındığında bulunan piksel yolunun çakışmadığını gözlemlemiş ve üst yarıya ait altı piksel yolu özelliği daha önermişlerdir [39]. Bu durum görüntünün bütününden bir piksel yolu çıkarıldığında, çıkarılan piksel yolu üst kısım için optimal olmadığından üst kısımda bozuklukların olacağını göstermektedir. Dolayısıyla, görüntünün üst yarısının piksel yolu özellikleri açısından incelenmesi piksel yolu çıkarma tabanlı yapılan değişiklikler hakkında ipucu verebilir.

Önerilen hibrit yöntem şu şekilde çalışmaktadır:

1. Eğitim ve test kümelerini oluştur.
2. Veri kümesindeki her görüntünün YİÖ dönüşümünü elde et.
3. Her görüntüden 24 YİÖ tabanlı ve 18 piksel yoğunluğu tabanlı özellik çıkar.
4. Görüntünün alt yarısından 12 piksel yolu özelliği çıkar (6 YİÖ tabanlı, 6 piksel yoğunluğu tabanlı).
5. Görüntünün üst yarısından piksel yoğunluğu tabanlı 6 piksel yolu özelliği çıkar.
6. DVM'yi 60 özellekle eğit.
7. Eğitilen DVM'yi eğitimde kullanılmayan görüntülerle test et.

### 3.1.1. Çıkarılan özellikler

Çıkarılan özellikler enerji tabanlı, piksel yolu tabanlı, gürültü tabanlı, üst yarı piksel yolu tabanlı ve alt yarı piksel yolu tabanlı olarak gruplanabilir. DVM'nin eğitiminde görüntüleri tanımlayan bazı istatistiksel özellikler kullanılmıştır. Bu özellikler hem piksel domeninde hem de YİÖ domeninde elde edilir. Aşağıdaki tablolarda bahsedilen özellikler  $n \times m$  boyutlu bir görüntü için formül olarak verilmiştir (Tablo 3.1., Tablo 3.2. ve Tablo 3.3.). Piksel yolu tabanlı özellikler KMEM (K) üzerinden hem satır hem sütun yönünde hesaplanır. Alt ve üst yarı piksel yolu özellikleri alt ve üst yarı görüntüdeki muhtemel tüm piksel yollarının yalnızca minimum, maksimum ve ortalama değerleridir. Görüntünün gürültüsü (G)  $5 \times 5$  pencereyi Wiener filtresi yardımıyla çıkarılır.

Tablo 3.1. Enerji tabanlı özellikler [32]

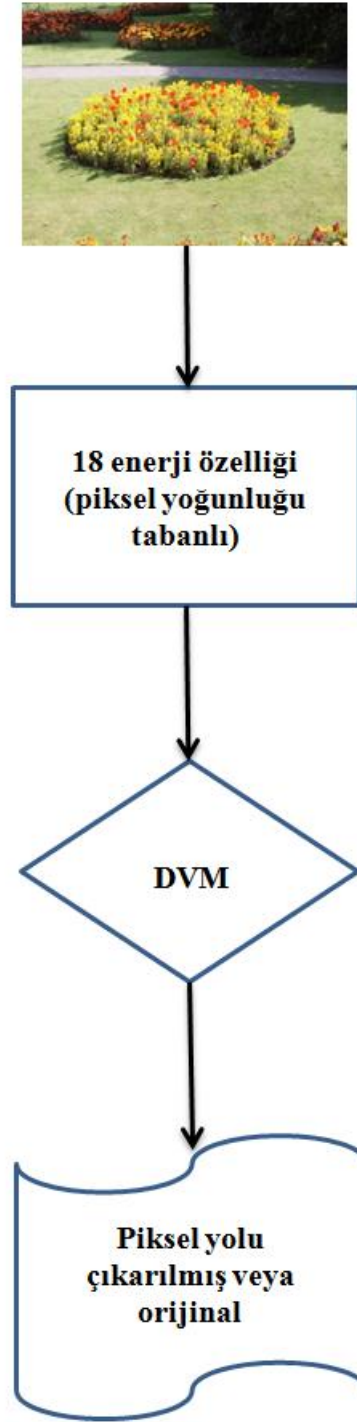
Özellik	Tanım
Ortalama satır enerjisi	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left  \frac{\partial}{\partial x} I(i, j) \right $
Ortalama sütun enerjisi	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left  \frac{\partial}{\partial y} I(i, j) \right $
Ortalama enerji	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \left  \frac{\partial}{\partial x} I(i, j) \right  + \left  \frac{\partial}{\partial y} I(i, j) \right  \right)$
Ortalama enerji farkı	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \left  \frac{\partial}{\partial x} I(i, j) \right  - \left  \frac{\partial}{\partial y} I(i, j) \right  \right)$

Tablo 3.2. Piksel yolu tabanlı özellikler [32]

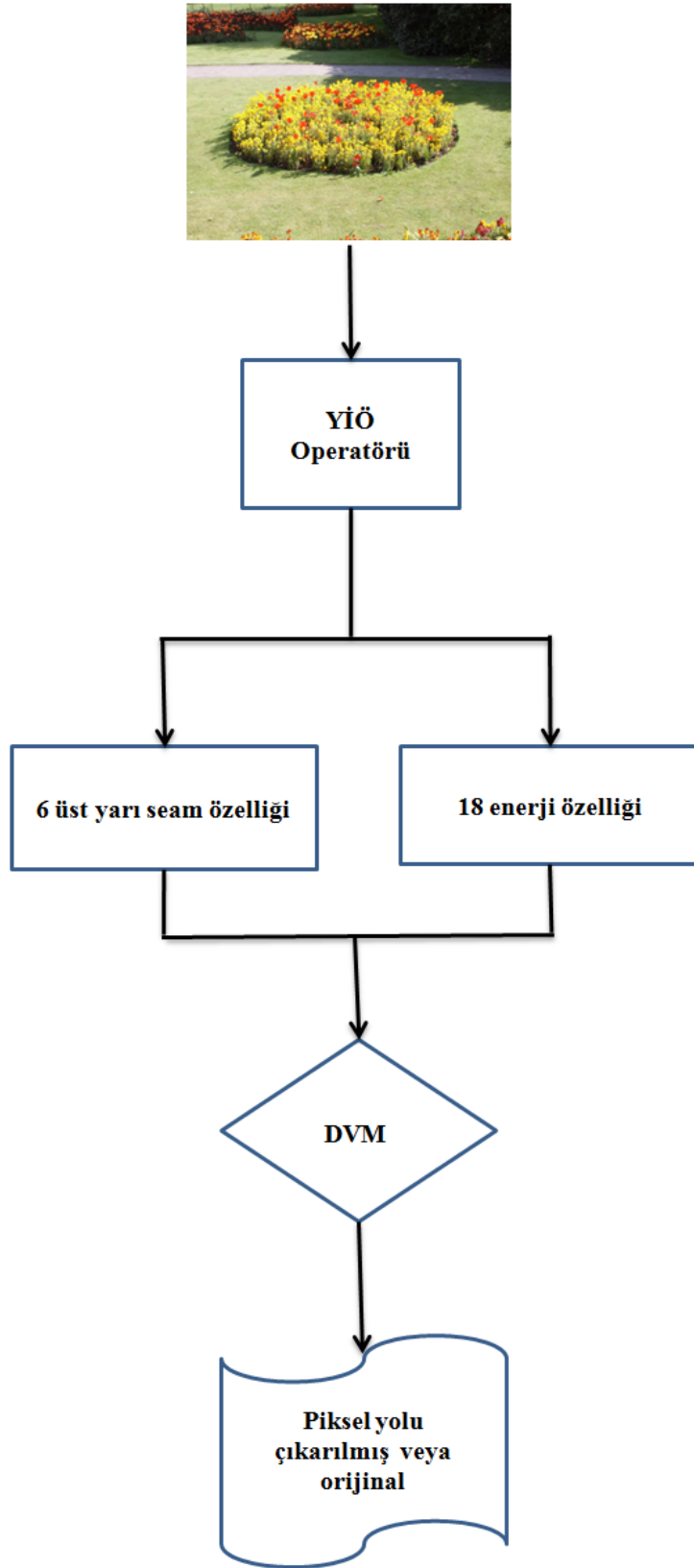
Özellik	Tanım
Maksimum	$\max_{i=1}^m K(i, n)$
Minimum	$\min_{i=1}^m K(i, n)$
Ortalama	$\frac{1}{m} \sum_{i=1}^m K(i, n)$
Standart Sapma	$\sqrt{\frac{1}{m} \sum_{i=1}^m (Ortalama - K(i, n))^2}$
Fark	$\max_{i=1}^m K(i, n) - \min_{i=1}^m K(i, n)$

Tablo 3.3. Gürültü tabanlı özellikler [32]

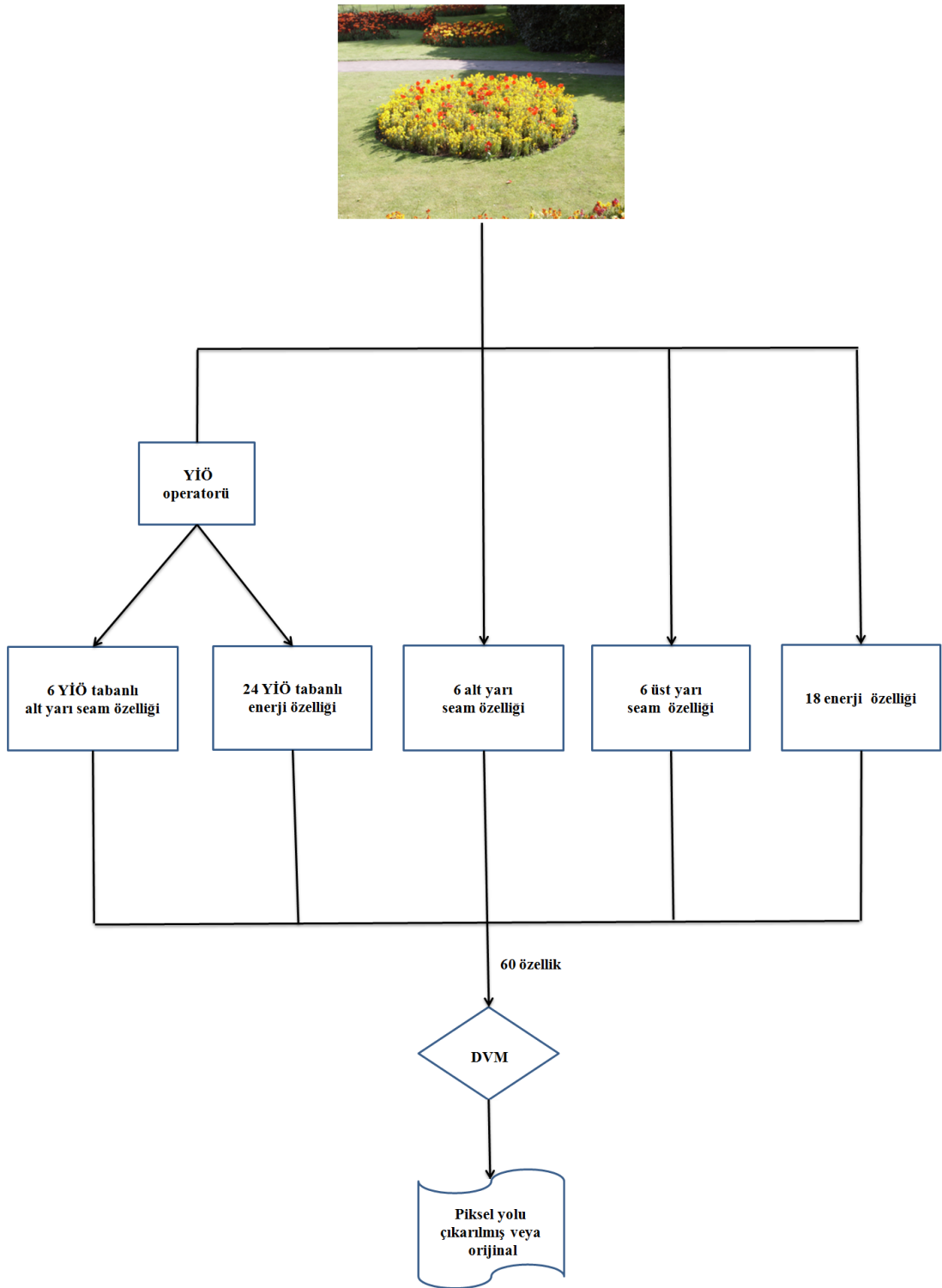
Özellik	Tanım
Ortalama	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n G(i, j)$
Standart Sapma	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (G(i, j) - Ortalama)^2$
Çarpıklık	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \frac{G(i, j) - Ortalama}{Standart Sapma} \right)^3$
Basıklık	$\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \left( \frac{G(i, j) - Ortalama}{Standart Sapma} \right)^4$



Şekil 3.2. Senaryo 1: 18 özellik [32] ile eğitim ve test



Şekil 3.3. Senaryo 2: 24 özellik [39] ile eğitim ve test



Şekil 3.4. Senaryo 3: 60 hibrit özellik ile eğitim ve test

### 3.2. Şerit Tabanlı Piksel Yolu Çıkarma Tespit Yöntemi

Literatürde enerji tabanlı, gürültü tabanlı ve yarı piksel yolu tabanlı özellikler de önerilmiş olmasına rağmen, piksel yolu çıkarma tespitinde en etkili özellikler piksel yolu tabanlı özelliklerdir [39]. Bu amaçla, bu tez çalışmasında eğitim ve test aşamalarında yalnızca piksel yolu tabanlı özellikler kullanılmıştır. Şerit tabanlı piksel yolu çıkarma tespit yöntemi, görüntüyü birkaç şeride böler ve her bir şeridi bahsedilen piksel yolu tabanlı özellikler açısından değerlendirir. Şeritlere ayırmanın arkasındaki mantık [39] ve [27]'den gelmektedir. Görüntünün bazı kısımlarında bulunan optimal piksel yolları arasındaki mesafe sıfırdan büyük olabilir. Bu da görüntüye piksel yolu çıkarma işlemi uygulandığını gösterir [27]. Yin ve ark. [39] şunu gözlemlemişlerdir: görüntünün üst yarısında bulunan optimal piksel yolu, alt yarıda bulunandan farklıdır. Bu amaçla eklenen üst yarı görüntüye ait özellikler tespit oranını artırmıştır. Buradan yola çıkarak, önerilen yeni piksel yolu çıkarma tespit yönteminde görüntülerin alt parçaları değerlendirilmiştir. Buna ek olarak, alt parçaların yalnızca tespit en etkili olan özellikleri, piksel yolu tabanlı özellikler, kullanılmıştır. Her şeritten onar piksel yolu tabanlı özellik çıkarılmıştır. Şekil 3.5.'de örnek olarak görüntü üzerinde dört şerit ve her şeritteki optimal piksel yolları ile bütün görüntünün optimal piksel yolu görülmektedir. Değerlendirilen tüm şeritler de Şekil 3.6.'da gösterilmiştir. Şekilde görüldüğü gibi görüntüler 1 şerit, 2 şerit, 4 şerit, 6 şerit ve 8 şerit olarak düşünülmüş ve her şeritteki yerel değişimler [32]'de bahsedilen 10 istatistiksel özellik açısından değerlendirilmiştir.  $n \times m$  boyutlarındaki bir görüntü için bu özellikler Tablo 3.4.'de verildiği gibidir.

Tablo 3.4.'de görülen özellikler KMEM'den elde edilebilecek muhtemel tüm piksel yollarının kümülatif enerji toplamları üzerinden hesaplanmaktadır. Bu da önerilen yaklaşımı diğerlerinden ayırır. Optimal piksel yolu üzerinden özellik hesabı yerine muhtemel tüm piksel yollarının özellik hesabında kullanılması, önerilen yaklaşımın kullanılan piksel yolu çıkarma yönteminden bağımsız, daha kararlı bir yöntem olmasını sağlamaktadır. Enerji haritasından (görüntünün gradyanı) KMEM oluşturulduktan sonra, KMEM'nin son satırının her elemanı muhtemel bir piksel yolunun son elemanı olarak kabul edilmekte ve en üst satıra kadar KMEM her biri için gezilmektedir. Muhtemel tüm piksel yollarının kümülatif toplamlarının

hesaplanmasında kullanılan algoritma Şekil 3.7.'de verilmiştir. Yöntem görsel olarak da Şekil 4.2.'de gösterilmektedir. Dolayısıyla,  $n \times m$  boyutlarındaki bir görüntü için  $n$  tane muhtemel piksel yolu bulunur. Tablo 3.4.'deki  $K_{ys}$  ve  $K_{ds}$  şeridin sırasıyla yatay ve dikey piksel yolları için KMEM'sini ifade etmektedir.  $K_{ds}$  2.1. 'de anlatıldığı gibi hesaplanır,  $K_{ys}$  ise görüntünün 90 derece döndürülmüş şekli üzerinden hesaplanmaktadır. Minimum enerji (yatay ve dikey yönlerde minimum toplam piksel yolu enerjisi), maksimum enerji (yatay ve dikey yönlerde maksimum toplam piksel yolu enerjisi), her iki yöndeki ortalama ve standart sapma ile minimum ve maksimum arasındaki fark her bir şerit için hesaplanır. Özellikler görüntülerin YİÖ dönüşümleri üzerinden hesaplanmaktadır, böylece yerel bozukluklar belirginleştirilmiştir [39]. 210 boyutlu bir özellik vektörü her bir görüntü için hesaplanmıştır. Oluşturulan bu vektörler DVM'nin girdisini oluşturmaktadır. Yapılan deneylerin akış şeması Şekil 3.8.'de verilmiştir.

Tablo 3.4. Bir şeridin piksel yolu tabanlı özellikleri

Özellik	Tanım
Şerit_Dikey Seam <sub>maks</sub>	$\max_{i=1}^m K_{ds}(i, n)$
Şerit_Dikey Seam <sub>min</sub>	$\min_{i=1}^m K_{ds}(i, n)$
Şerit_Dikey Seam <sub>ortalama</sub>	$\frac{1}{m} \sum_{i=1}^m K_{ds}(i, n)$
Şerit_Dikey Seam <sub>std</sub>	$\sqrt{\frac{1}{m} \sum_{i=1}^m (Ortalama - K_{ds}(i, n))^2}$
Şerit_Dikey Seam <sub>fark</sub>	$\max_{i=1}^m K_{ds}(i, n) - \min_{i=1}^m K_{ds}(i, n)$
Şerit_Yatay Seam <sub>maks</sub>	$\max_{i=1}^n K_{ys}(m, i)$
Şerit_Yatay Seam <sub>min</sub>	$\min_{i=1}^n K_{ys}(m, i)$
Şerit_Yatay Seam <sub>ortalama</sub>	$\frac{1}{n} \sum_{i=1}^n K_{ys}(m, i)$
Şerit_Yatay Seam <sub>std</sub>	$\sqrt{\frac{1}{n} \sum_{i=1}^n (Ortalama - K_{ys}(m, i))^2}$
Şerit_Yatay Seam <sub>fark</sub>	$\max_{i=1}^n K_{ys}(m, i) - \min_{i=1}^n K_{ys}(m, i)$





Şekil 3.5. Orijinal görüntünün optimal piksel yolu (üstte), dört şerit ve her şeridin optimal piksel yolu (altta)



Şekil 3.6. Pikel yolu çıkarma tespiti için değerlendirilen şeritler

---

**Algorithm 1: Muhtemel her seam için kumulatif toplamlar**

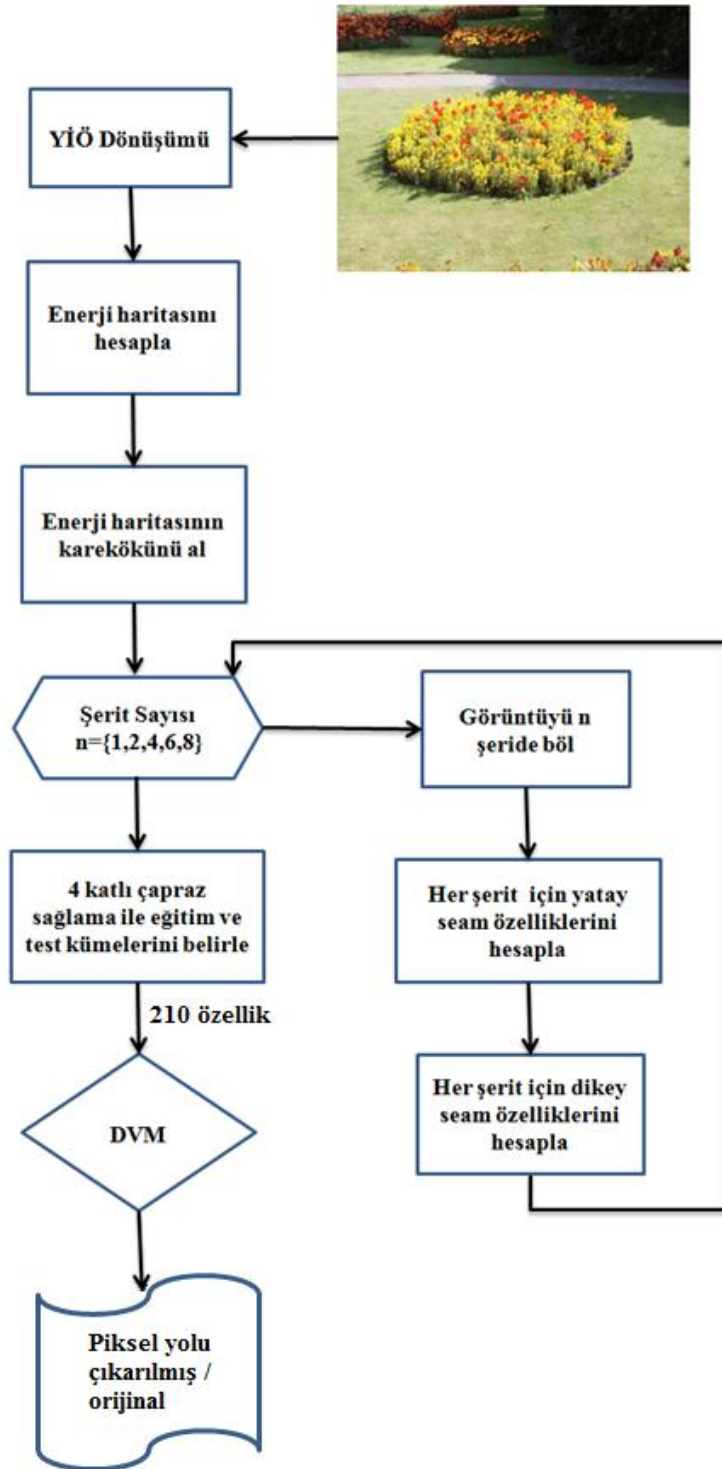

---

**Input:** G: YIO görüntüsünün gradyanı  
**Output:** M: Muhtemel seam yollarının toplami

- 1 Her muhtemel seam için KMEM'yi hesapla
- 2  $[nRows, nCols] \leftarrow size(G)$
- 3 for  $i \leftarrow 1$  to  $nRows$  do
- 4     for  $j \leftarrow 1$  to  $nCols$  do
- 5         if  $j=0$  then
- 6              $KMEM(i, j) = KMEM(i, j) + (KMEM(i, j) <$   
                $KMEM(i-1, j+1)) ? KMEM(i-1, j) : KMEM(i-1, j+1)$
- 7         else if  $j=nCols-1$  then
- 8              $KMEM(i, j) = KMEM(i, j) + (KMEM(i-1, j-1) <$   
                $KMEM(i-1, j)) ? KMEM(i-1, j-1) : KMEM(i-1, j)$
- 9         else
- 10              $temp \leftarrow (KMEM(i-1, j-1) <$   
                $KMEM(i-1, j)) ? KMEM(i-1, j-1) : KMEM(i-1, j)$
- 11              $temp \leftarrow (temp < KMEM(i-1, j+1)) ? temp : KMEM(i-1, j+1)$
- 12              $KMEM(i, j) \leftarrow KMEM(i, j) + temp$
- 13 Muhtemel her seam için kumulatif enerji toplamlarını (M) hesapla
- 14 for  $i \leftarrow 1$  to  $nCols$  do
- 15      $pix(end) \leftarrow i$
- 16     for  $k \leftarrow nRows - 1$  to 1 do
- 17          $m \leftarrow \min(KMEM(k, \max(pix(k+1) - 1, 1) : \min(pix(k+1) + 1, end)))$
- 18          $pix(k) \leftarrow m + pix(k+1) - 1 - (pix(k+1) > 1)$
- 19     Seam yollarının toplamını hesapla
- 20      $tmp \leftarrow 0$
- 21     for  $k \leftarrow 1$  to  $m$  do
- 22          $tmp \leftarrow tmp + KMEM(k, pix(k))$
- 23      $M(i) \leftarrow tmp$
- 24 return M

---

Şekil 3.7. Muhtemel tüm piksel yollarının kümülatif toplamlarını hesaplayan algoritma



Şekil 3.8. Şerit tabanlı yöntemin akış şeması

## BÖLÜM 4. DENEYSEL SONUÇLAR VE TARTIŞMA

### 4.1. Deneysel Altyapı

Önerilen piksel yolu çıkarma tespit yöntemi, özellikle sayısal hesaplamalar için geliştirilmiş bir yazılım olan Matlab (TM) üzerinde DVM sınıflayıcılar LibSVM [90] ve özellikle karışık testlerdeki yoğun aramalar için cuSVM kütüphaneleri kullanılarak gerçekleştirilmiştir [72]. Çekirdek fonksiyonu olarak Radyal Temelli Fonksiyon kullanılmıştır. Bu fonksiyonun parametrelerinin ( $c$  ve  $\gamma$ ) optimal değerleri ızgara arama yöntemi ile belirlenmiştir. Ayrıca, tespit yaklaşımının eğitim ve test aşamalarında kullanılmak üzere bir veritabanı oluşturulmuştur. Bu amaçla UCID [30] veritabanından alınan 1338 orijinal görüntü piksel yolu çıkarma ile %3, %6, %10, %20 ve %30 ölçeklenmiştir. UCID veritabanındaki bazı görüntüler Şekil 4.1.'de örnek olarak gösterilmiştir. Oluşturulan veritabanında 1338 orijinal ve  $1338 \times 5 = 6690$  işlenmiş görüntü (toplam 8028 görüntü) bulunmaktadır. Yapılan tüm deneylerde piksel yolu çıkarma görüntünün genişliğinin azaltılması şeklinde uygulanmıştır. Önceki yöntemlerle ([32], [39]) eşit karşılaştırma yapabilmek için eğitim ve testte aynı ölçekleme oranları kullanılmıştır. Örneğin, %30 ölçeklemenin tespitinde yalnızca orijinal görüntülerle %30 işlenmiş görüntüler kullanılmıştır. Dört katlı çapraz sağlama yöntemi kullanılmıştır. Veri kümesinin rastgele seçilen dörtte üçlük kısmı eğitimde dörtte birlik kısmı testte kullanılmıştır. Tespit yönteminde çıkarılan özellikler KMEM'den elde edilebilecek muhtemel tüm piksel yolları üzerinden hesaplanmıştır. Şekil 4.2'de KMEM'deki muhtemel piksel yolları farklı renklerle gösterilmiştir. Tüm hesaplamalar Intel Core i7-4720HQ CPU @ 2.60 GHz, 16 GB RAM ve Nvidia GeForce GTX960M grafik işlemciye sahip kişisel bilgisayar üzerinde gerçekleştirilmiştir.



Şekil 4.1. UCID veritabanından örnek görüntüler

6	14	4	12	11	25	21	15
42	32	51	39	55	62	95	18
76	58	40	42	56	71	25	21
62	45	60	50	45	29	31	62
47	49	49	50	39	37	65	35
55	52	75	67	58	57	56	95
68	70	82	78	62	59	74	111
75	92	91	83	69	77	82	75
90	84	115	93	89	88	87	144

Şekil 4.2. KMEM'deki muhtemel piksel yolları

#### 4.2. Hibrit Yöntem İle Tespit Sonuçları

Deneyle Şekil 3.2., Şekil 3.3. ve Şekil 3.4.'de verilen senaryolara göre yapılmıştır. Kısacası, ilk başta [32]'daki yöntemin tespit oranları elde edilmiştir. 18 enerji tabanlı özellik dört katlı çapraz sağlama ile oluşturulan eğitim ve test kümeleri ile DVM'ye verilmiştir. Daha sonra YİÖ dönüşümünün etkisi gösterilmiştir. Görüntülerin YİÖ dönüşümleri yapıldıktan sonra bahsedilen 18 özellik ve görüntünün üst yarısına ait ek 6 özellik çıkarılmıştır [39]. Son olarak, önerilen hibrit yöntem uygulanmıştır. 6 üst yarı ve 6 alt yarı özellik 18 enerji tabanlı özelliğe eklenmiştir. Benzer şekilde YİÖ tabanlı alt yarı özellikleri de eklenmiştir. Elde edilen bu hibrit kümede 60 adet YİÖ ve piksel yoğunluğu tabanlı özellik bulunmaktadır. Hibrit küme ile elde edilen tespit doğrulukları Tablo 4.1.'de verilmiştir.

Tablo 4.1. Hibrit yaklaşım ile tespit doğrulukları (%)

Ölçekleme Oranı	[32]	[39]	Hibrit
3%	53,25	54,26	<b>61,85</b>
6%	63,19	64,57	<b>73,02</b>
10%	71,53	81,73	<b>85,69</b>
20%	79,56	93,05	<b>94,69</b>
30%	86,40	96,26	<b>97,65</b>
Ortalama	70,79	77,97	<b>82,58</b>

Tablo 4.2. Hibrit yöntemin Yanlış Pozitif Oranı (%)

Ölçekleme Oranı	[32]	[39]	Hibrit
3%	44,91	57,91	<b>41,19</b>
6%	33,23	45,07	<b>31,34</b>
10%	21,26	26,95	<b>19,16</b>
20%	17,07	8,383	<b>7,16</b>
30%	10,75	5,090	<b>3,88</b>
Ortalama	25,44	28,68	<b>20,55</b>

Tablo 4.3. Hibrit yöntemin Yanlış Negatif Oranı (%)

Ölçekleme Oranı	[32]	[39]	Hibrit
3%	49,10	31,44	<b>29,55</b>
6%	41,02	28,14	<b>21,86</b>
10%	30,24	13,47	<b>9,55</b>
20%	18,51	8,68	<b>3,28</b>
30%	16,42	2,69	<b>1,79</b>
Ortalama	31,06	16,88	<b>13,21</b>

### 4.3. Şerit Tabanlı Yöntem İle Tespit Sonuçları

Şekil 3.8.'de verilen akış şeması, önerilen orijinalden bağımsız (blind) tespit yöntemini özetlemektedir. Veri tabanındaki tüm orijinal ve piksel yolu çıkarılmış görüntüler YİÖ domenine dönüştürülür. Özellik çıkarımı enerji haritasının karekökü üzerinden gerçekleştirilir. Enerji haritasının karekökünü almak veri aralığını kısıtlar ve özellikleri daha belirleyici yapar. Görüntüler 1 şerit, 2 şerit, 4 şerit, 6 şerit ve 8 şerit halinde ele alınır. Her şerit yukarıda bahsedilen 10 istatistiksel özellik açısından değerlendirilir [32]. Her görüntüden toplamda 210 özellik çıkarılır ve  $8028 \times 210$  boyutlu özellik matrisi oluşturulur. Dört katlı çapraz sağlama yöntemi eğitim ve test



kümelerini belirler ve deneylerin rastgeleliğini garantiler. Test görüntüleri eğitim görüntülerinden farklıdır.

Tespit için çıkarılan tüm özellikler piksel yolu davranışı ile alakalıdır. Piksel yolu çıkarma/ekleme algoritması boyut azaltma ya da artırmada piksel yolu çıkarıp eklediğinden, görüntüde bu işlemde en çok etkilenen öge piksel yollarıdır. Piksel yollarının ortalama enerjisi piksel yolu çıkarıldığında artar, dolayısıyla enerji değerlendirmeleri, yapılan boyutlama işlemini ele verir. YİÖ tabanlı piksel yolu çıkarma tespit yönteminde de en etkili özellikler piksel yolu tabanlı özellikler olduğundan [39] önerilen orijinalden bağımsız (blind) tespit yönteminde yalnızca bu özellikler kullanılmıştır. Önerilen şerit tabanlı yöntemin her ölçekleme oranına göre tespit performansı Tablo 4.4.'de verilmiştir. Elde edilen tespit doğrulukları literatürdeki en yüksek tespit oranlarına göre %20'ye kadar artırılmıştır. Bu gelişim, YİÖ'den yararlanarak görüntünün alt parçalarının incelenmesi sonucu ortaya çıkmıştır. Bir görüntünün optimal piksel yolunun genellikle o görüntünün alt parçalarının (şeritler) optimal piksel yolu ile aynı olmadığı gözlemlenmiştir ([39] ile uyumlu bir çıkarım). Dolayısıyla, şeritler için piksel yolu çıkarma süreci ile alakalı istatistiksel özellikleri detaylı olarak analiz etmek piksel yolu çıkarma tabanlı görüntü ölçeklemenin tespitine yardım etmiştir.

Tablo 4.4. Tespit doğrulukları (%)

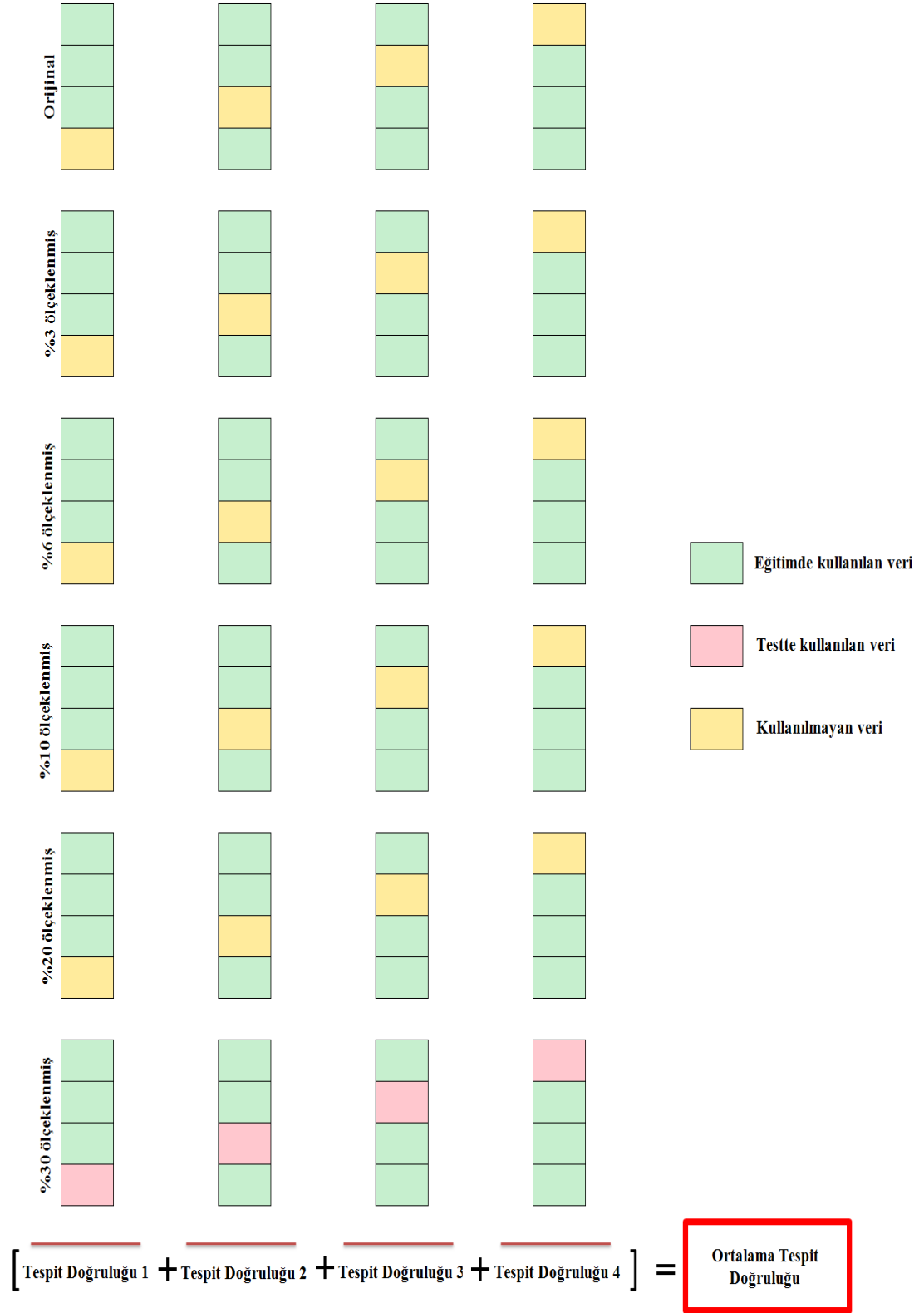
Ölçekleme Oranı	[32]	[39]	Şerit tabanlı
3%	53,25	55,64	<b>74,48</b>
6%	63,19	66,63	<b>87,52</b>
10%	71,53	78,31	<b>94,02</b>
20%	79,56	91,52	<b>98,54</b>
30%	86,40	95,43	<b>99,55</b>
Ortalama	70,79	77,51	<b>90,82</b>

Ayrıca, daha gerçekçi tespit oranlarını görebilmek için şerit tabanlı yöntemin karışık veri kümesi ile testi de gerçekleştirilmiştir. Gerçek dünyadaki adli analizlerde ölçekleme oranı bilinmeyeceğinden karışık veri kümesi ile yapılan deneyler önerilen şerit tabanlı yöntemin gerçekçi sonuçlarını vermektedir. Karışık veri kümesi ile test görüntününün %30 ölçeklenip ölçeklenmediğinin tespiti Şekil 4.3.'de görsel

olarak gösterilmiştir. Bu amaçla orijinal görüntüler ve piksel yolu çıkarma yöntemi ile %3, %6, %10, %20, %30 oranlarında ölçeklenmiş görüntüler DVM'ye girdi olarak verilmiştir. Şerit tabanlı yöntemin performansı bu şekilde ölçülmüştür. Sonuçlar Tablo 4.5.'de gösterilmektedir. Sonuçlara göre, karışık testlerde tespit oranında beklenen bir düşüş olsa da şerit tabanlı yöntem, mevcut en başarılı tespit yöntemini ölçekleme oranları karışık olan veri kümesi ile ortalamada yaklaşık %12 gelişim ile geride bırakmaktadır.

Tablo 4.5. Karışık veri kümesi ile tespit doğrulukları (%)

Ölçekleme Oranı	[32]	[39]	Şerit tabanlı
3%	52,51	55,33	<b>71,63</b>
6%	53,72	66,03	<b>81,84</b>
10%	61,76	73,89	<b>87,32</b>
20%	64,58	82,25	<b>89,79</b>
30%	68,97	85,16	<b>91,57</b>
Ortalama	60,31	72,53	<b>84,43</b>



Şekil 4.3. %30 ölçeklemenin karışık veri kümesi ile testi

Buna ek olarak, görüntüyü şeritlere bölmenin etkisi de incelenmiştir. Bu amaçla beş deney daha yapılmıştır. İlkinde, görüntünün bütünü bir şerit olarak düşünülmüş ve on özellik çıkarılmıştır. Yalnızca bu on özellik DVM'ye gidi olarak verilmiş ve performans ölçülmüştür. Daha sonra, görüntü iki şerit olarak düşünülmüş ve her bir şeritten on tane olmak üzere toplamda yirmi özellik DVM'ye girdi olarak verilerek performans ölçülmüştür. Diğer üç değerlendirme de benzer şekilde yapılmıştır. Tablo 4.6. şerit sayılarına göre yöntemin tespit doğruluklarını göstermektedir. Bu tablodan görüntüyü altı şeride bölerek  $6 \times 10 = 60$  özellik çıkarmak ve DVM'yi bu 60 özellekle eğitmenin en iyi tespit doğruluklarını verdiği görülmektedir. Altı şeridin değerlendirmeye katılması her ölçekleme oranında en iyi sonuçları üretmiştir. Bu da kullanılan veri kümesine göre optimal şerit derinliğinin altı olduğunu göstermektedir. Altı şeritten az ya da fazla şerit ile tespit oranları düşmektedir. Altı şeridin başarımlı karışık veri kümesi ile de test edilmiş ve yine en yüksek tespit oranları elde edilmiştir (Tablo 4.7.). Yöntem genel itibarıyla literatürdeki diğer tespit yöntemlerine ciddi fark atmaktadır (Tablo 4.8.).

Tablo 4.6. Şerit sayısına göre tespit doğrulukları (%)

Ölçekleme Oranı	1 Şerit	2 Şerit	4 Şerit	6 Şerit	8 Şerit
3%	61,66	67,60	74,29	<b>77,43</b>	74,89
6%	73,40	84,12	89,76	<b>92,04</b>	86,17
10%	82,66	93,09	96,97	<b>97,68</b>	94,10
20%	92,15	98,43	99,14	<b>99,66</b>	98,84
30%	96,45	99,33	99,88	<b>99,93</b>	99,63
Ortalama	81,26	88,51	92,01	<b>93,35</b>	90,73

Tablo 4.7. 6 şerit ve karışık veri kümesi ile tespit doğrulukları (%)

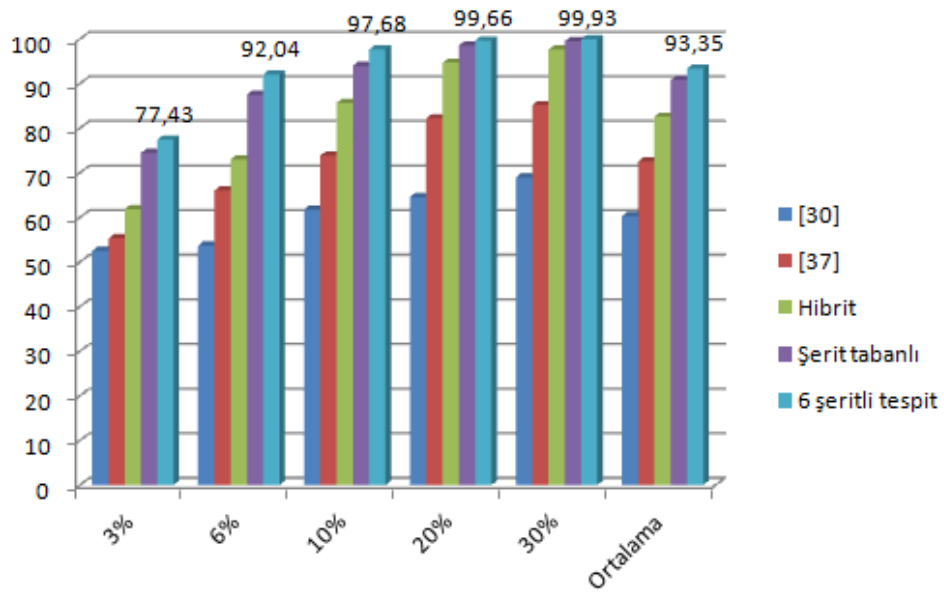
Ölçekleme Oranı	[32]	[39]	6 şerit ile tespit
3%	52,51	55,33	<b>75,13</b>
6%	53,72	66,03	<b>89,23</b>
10%	61,76	73,89	<b>93,66</b>
20%	64,58	82,25	<b>95,23</b>
30%	68,97	85,16	<b>97,13</b>
Ortalama	60,31	72,53	<b>90,08</b>

Tablo 4.8. 6 şerit ile tespitin diğer yöntemlerle tespit doğrulukları (%) açısından karşılaştırılması

Ölçekleme Oranı	[32]	[39]	6 şeritli tespit	Fark
3%	52,51	55,33	<b>77,43</b>	22,10
6%	53,72	66,03	<b>92,04</b>	26,01
10%	61,76	73,89	<b>97,68</b>	23,79
20%	64,58	82,25	<b>99,66</b>	17,41
30%	68,97	85,16	<b>99,93</b>	14,77
Ortalama	60,31	72,53	<b>93,35</b>	20,82

Tablo 4.9. Tüm yöntemlerin başarımları ve 6 şeritli yöntemin [39] ile farkı

Ölçekleme Oranı	[32]	[39]	Hibrit	Şerit tabanlı	6 şeritli tespit	Fark
3%	52,51	55,33	<b>61,85</b>	<b>74,48</b>	<b>77,43</b>	22,10
6%	53,72	66,03	<b>73,02</b>	<b>87,52</b>	<b>92,04</b>	26,01
10%	61,76	73,89	<b>85,69</b>	<b>94,02</b>	<b>97,68</b>	23,79
20%	64,58	82,25	<b>94,69</b>	<b>98,54</b>	<b>99,66</b>	17,41
30%	68,97	85,16	<b>97,65</b>	<b>99,55</b>	<b>99,93</b>	14,77
Ortalama	60,31	72,53	<b>82,58</b>	<b>90,82</b>	<b>93,35</b>	20,82

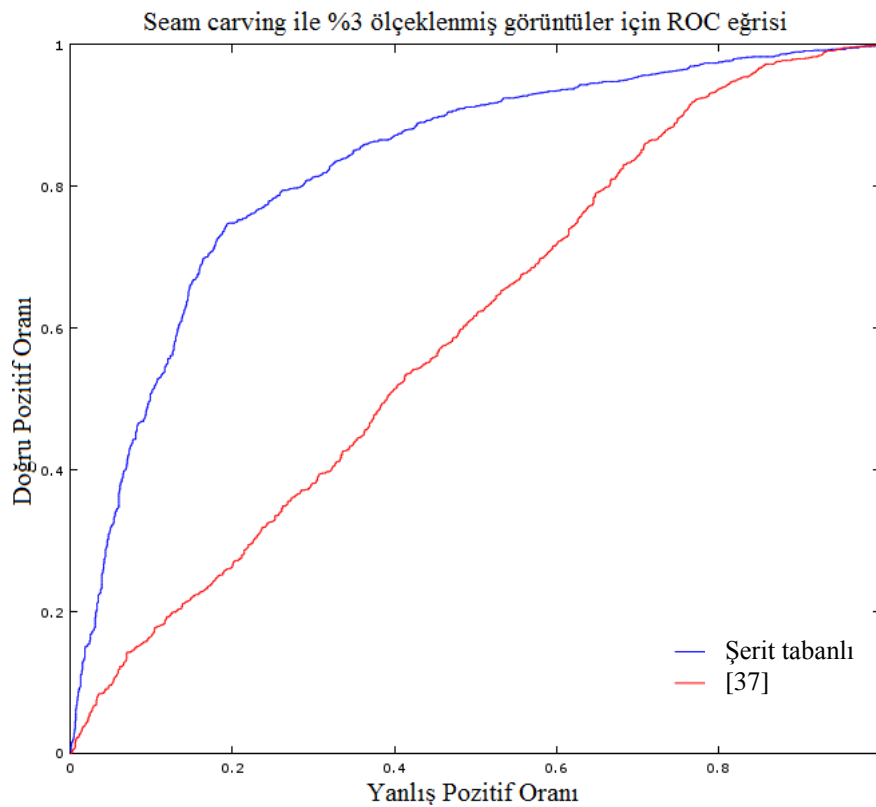


Şekil 4.4. Tüm yöntemlerin tespit doğruluklarının karşılaştırılması

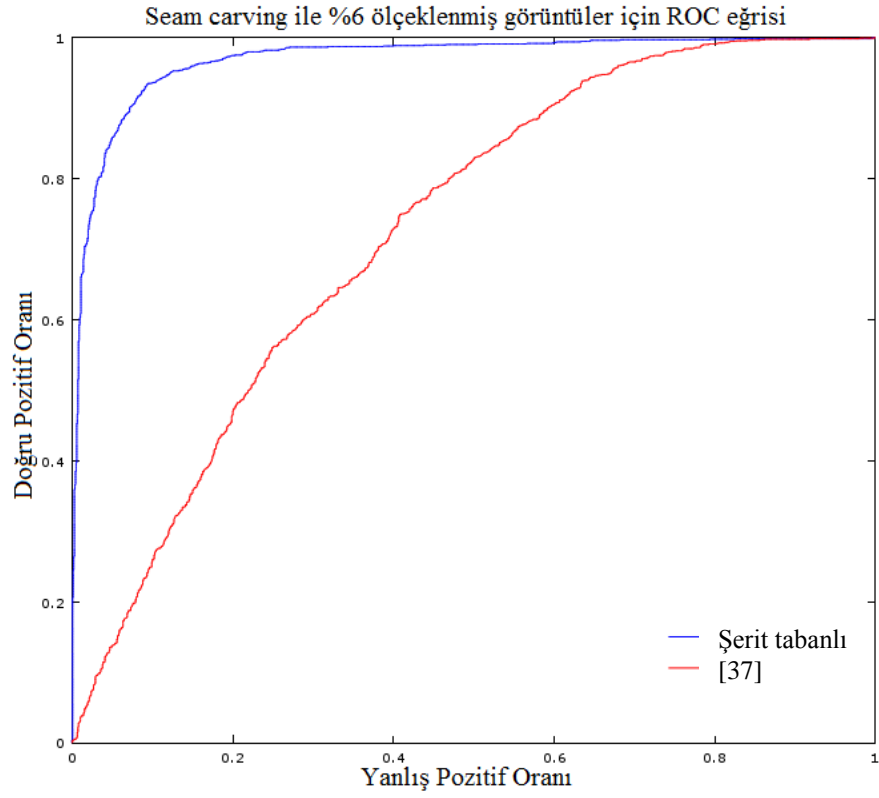
Tablo 4.9.'da geliştirilen tüm yöntemler ve literatürdeki en başarılı iki yöntemin ölçekleme oranlarına göre tespit doğrulukları verilmiştir. Kat edilen gelişim kayda değerdir. Şekil 4.4.'de da tüm yöntemlerin başarımları grafiksel olarak ifade edilmiştir. 6

şeritli tespit yöntemi her ölçekleme oranında en yüksek performansı sergileyerek önerilen en başarılı yöntem olmuştur. Şerit tabanlı yönteminin daha iyi ispatı için her ölçekleme oranına göre ROC (Receiver Operating Characteristics) analizi yapılmıştır. Şekil 4.5. - Şekil 4.9.'da literatürdeki en başarılı yöntem [39] ile şerit tabanlı yönteminin ROC eğrileri karşılaştırılmıştır. Karşılaştırma için doğru pozitif ve yanlış pozitif sayıları hesaplanmıştır. ROC eğrileri incelendiğinde önerilen yöntemin başarısı açıkça görülmektedir. Özellikle tespitin çok daha zor olduğu düşük ölçekleme oranlarında başarı önemli ölçüde geliştirilmiştir.

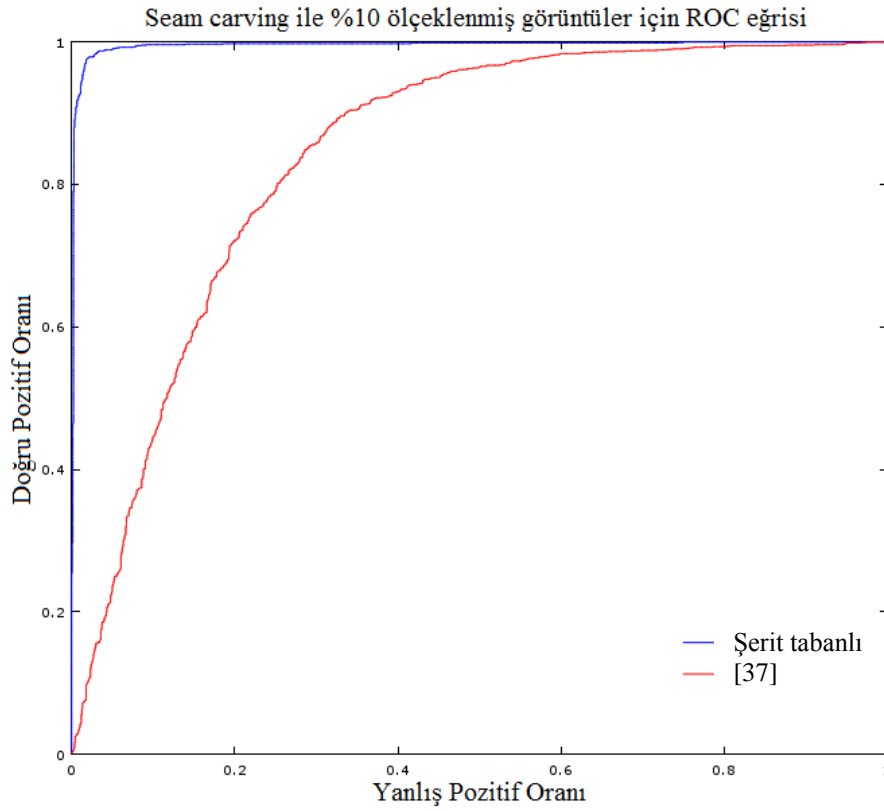
Tablo 4.10., doğru pozitif oranı (DPO) ve yanlış pozitif oranını (YPO) her ölçekleme oranı için göstermektedir. Tablodan açıkça görülmektedir ki, şerit tabanlı yöntem DPO'yu diğer yöntem [39] göre daha az YPO ile ortalama %17 artırmıştır. Sebebi, görüntünün alt parçalarının (şeritlerin) daha etkili özelliklerle (piksel yolu tabanlı özelliklerle) incelenmesidir. Önerilen tespit yaklaşımı ile YPO ve YNO geçmiş yöntemlere göre önemli ölçüde düşürülmüştür. Bu da yanlış sınıflandırılan örneklerin sayısının önceki yöntemlere göre oldukça az olduğunu göstermektedir.



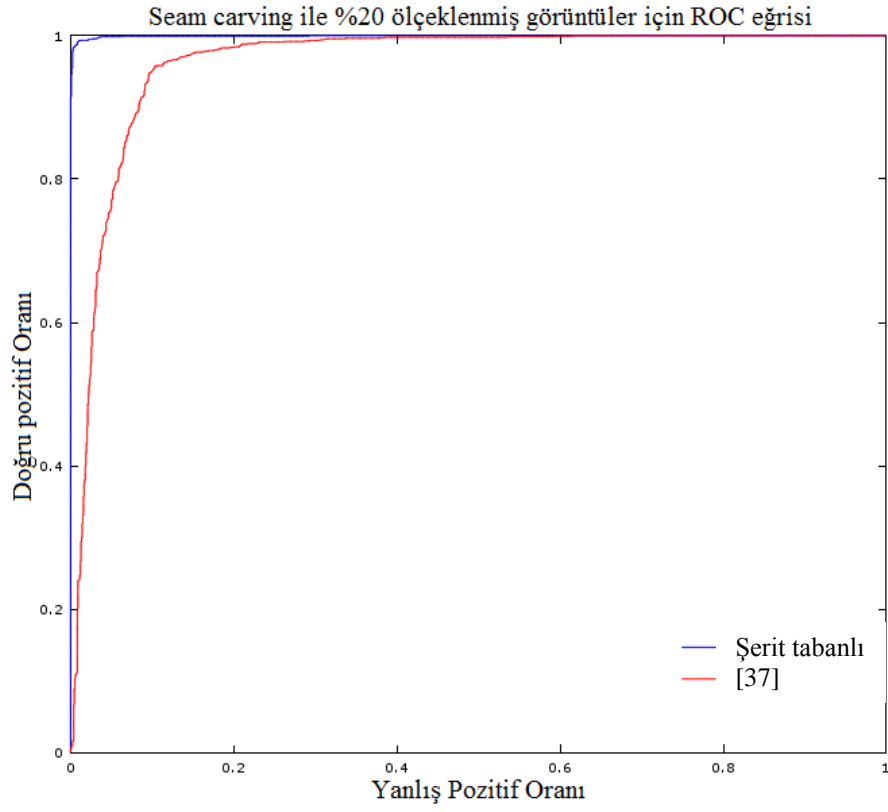
Şekil 4.5. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%3 için)



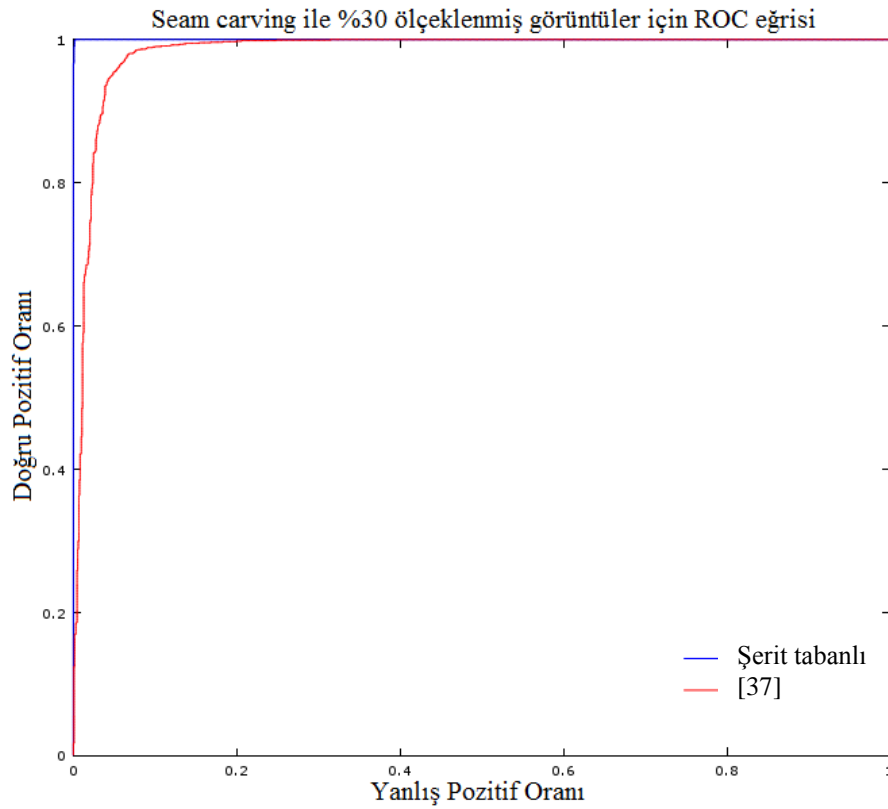
Şekil 4.6. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%6 için)



Şekil 4.7. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%10 için)



Şekil 4.8. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%20 için)



Şekil 4.9. Şerit tabanlı yöntem ile literatürdeki en başarılı yöntemin ROC eğrilerinin karşılaştırılması (%30 için)



Tablo 4.10. ROC analizi

Ölçekleme Oranı	[37]		6 şeritli tespit	
	DPO	YPO	DPO	YPO
3%	0,55	0,41	0,77	0,24
6%	0,64	0,33	0,86	0,10
10%	0,80	0,23	0,96	0,06
20%	0,89	0,13	0,98	0,01
30%	0,94	0,07	0,99	0,00
Ortalama	0,76	0,23	0,91	0,08

#### 4.4. Tartışma

Denenen hibrit yöntemle bazı ek özellikler sayesinde YİÖ tabanlı seam carving tespit yöntemi iyileştirilmiştir. Yöntem, YİÖ tabanlı ve piksel yoğunluğu tabanlı özellikleri birleştirerek hibrit bir yaklaşım sunar. Kat edilen gelişimin sebepleri şunlardır: piksel domeninde alt ve üst yarı seam özellikleri eklemek seam carving sonrası enerjideki değişimi ortaya çıkarır. Seam carving az enerjili seam'leri çıkardığından görüntünün ortalama enerjisi artar. Alt ve üst yaruları incelemenin mantığı [39] ile aynıdır. Yani, görüntünün üst ve alt yarısında bulunan seam'ler birbirinden farklıdır. Bir yer için optimal olan diğer yer için optimal değildir. Öyle olunca da görüntünün ilgili kısmında enerji dengesi zarar görür. Dikey bir seam de alttan üste doğru bulunabileceği gibi üstten alta doğru da bulunabilir. Dolayısıyla, üst yarı özelliklerin yanında alt yarı özellikleri de çıkarmak tespit oranını artırır. Tablo 4.1.'den görüldüğü gibi yeni önerilen özelliklerin etkisi düşük ölçekleme oranlarında (%3 ve %6) kendini daha çok belli etmektedir. Az miktarda seam carving çok miktarda enerji kalması demektir. Dolayısıyla, küçük ölçekleme oranlarında ölçeklenen görüntülerde enerji ile ilgili özellikler daha yüksek olacaktır. Ayrıca bu görüntülerdeki muhtemel tüm seam'ler daha yüksek enerji taşıyacaktır, seam özelliklerinin değerleri daha yüksek olacaktır. Bunun yanında, yüksek ölçekleme oranlarında daha fazla düz alan çıkarılacağından gürültü seviyesi etkilenecek ve görüntünün gürültüsü ile alakalı özellikler bu durumu yansıtacaktır. Gürültü özelliklerinin etkisi yüksek ölçekleme oranlarında daha belirgin olmaktadır; fakat bu etki enerji ve seam özellikleri tarafından bastırılmaktadır.

Tablo 4.2. ve Tablo 4.3.'de hibrit yöntemin sırasıyla Yanlış Pozitif Oranları (YPO) ve

Yanlış Negatif Oranları (YNO) gösterilmiştir. Yöntemin YPO ve YNO değerleri karşılaştırılan diğer yöntemlere göre oldukça azdır. Yani, yanlış sınıflandırılan örnek sayısı önceki yöntemlere göre oldukça azdır. YPO ortalama %8, YNO ise ortalama %3 iyileştirilmiştir.

Piksel yolu çıkarma tabanlı ölçeklemeyi DVM kullanarak tespit etmek birkaç orijinalden bağımsız (blind) tespit yöntemi arasında en yaygın olanıdır. Buradaki ana nokta DVM'nin girdilerinin (özelliklerin) layığıyla belirlenmesidir. Bu noktada, yedi yıldır piksel yolu çıkarma tespitinde kullanılan pek çok özellik çıkarımı yaklaşımı mevcuttur. En başarılı özellikler piksel yolu çıkarma metodolojisini kullanan özelliklerdir. Piksel yolu çıkarma sürecini irdelemek en yüksek doğruluk oranlarını vermiştir [32], [39]. Bu çıkarımların etkisini güçlendirmek için tespit sürecine yerel doku tanımlayıcı dâhil edilmiştir [39]. Bu katkı ile birlikte, piksel yolu çıkarmanın yerel bozukluklara sebep olması ve bu bozuklukların da yerel doku tanımlayıcı, YİÖ, ile belirginleştirilmesi sebebiyle tespit oranları geliştirilmiştir. Bunun da üstüne, görüntülerin parçalı olarak değerlendirilmesi, piksel yolu çıkarmanın görüntülerin detayındaki etkisini irdelemeye olanak sağlamıştır. Görüntünün alt kısımlarından bazı istatistiksel özelliklerin çıkarılması ve bunların DVM'ye verilmesi daha iyi tespit doğrulukları sağlamıştır. Bu gelişimin sebebi şudur: görüntü bir şeritler bütünü olarak düşünüldüğünde, dikey bir piksel yolu baştan aşağı tüm görüntüyü gezer ve piksel yolu çıkarma ya da eklemeyi tüm şeritler etkilenir. Bir piksel yolu çıkarıldığında en düşük enerjili piksel yolu çıkarılacağından şeritlerin ortalama enerjisi artar. Bu, bir şeridin genel enerji özelliklerinin değişmesi anlamına gelmektedir. Dolayısıyla her şeridin enerji tabanlı piksel yolu özelliklerini incelemek tespit performansına büyük katkı sağlar. Tablo 4.6., önerilen yöntemin şerit sayısına göre tespit doğruluklarını göstermektedir ve en yüksek oranlar altı şeritle elde edilmiştir. Altı şeritle tespit (60 özellik), 210 özellikli tespite göre daha başarılıdır. Bunun sebebi DVM'nin girdi sayısı ile alakalı olabilir. Fazla sayıda özellik tespit doğruluğunda azalmaya sebep olabilir. Nihayetinde, nokta sayısı arttığında sınıfları tam olarak ayıracak, tüm noktaları doğru sınıfa dâhil edecek hiper düzlemi elde etmek zorlaşacaktır. Deneylerde kullanılan veri tabanındaki görüntülere göre, piksel yolu çıkarma tabanlı görüntü boyutlandırmanın tespitinde ideal şerit sayısı altıdır şeklinde bir genelleme yapılabilir. Şerit sayısının görüntülerin çözünürlüğüne bağlı

olarak deęişebileceęini düşünmek doğaldır. Yüksek çözünürlüklü görüntülerde optimal derinlik artabilir, yani sekiz şerit yetmeyebilir. KMEM'den elde edilebilecek tüm muhtemel piksel yollarının özellik çıkarımında hesaba katılması (Şekil 4.2.) önerilen yöntemi daha kararlı bir hale getirmiştir. Yöntemdeki zorluk, zaman karmaşasıdır. Her bir şeritten onar istatistiksel özellięin çıkarılması ve DVM'nin bu özelliklerle eğitilmesi saatler almaktadır. Özellikle bir de karışık veri kümesi ile test ederken daha uzun sürmektedir. Ayrıca, DVM'nin optimal parametrelerine karar vermek de çok zaman harcamaktadır.

## BÖLÜM 5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında popüler içeriğe duyarlı görüntü boyutlama yöntemlerinden piksel yolu çıkarma yöntemi ile boyutları değiştirilmiş görüntülerin tespiti ele alınmıştır. Görüntülerin orijinalliğinin kesinleştirilmesi bazı durumlarda önemli olabilir. Klasik boyutlama yöntemlerinin aksine, piksel yolu çıkarılmış görüntülerin tespiti oldukça zordur. Bu zorluğun sebebi, görüntülerin geometrik özelliklerinin dışında, piksellerin önem derecelerinin de değerlendiriliyor olmasıdır. Bunun için daha detaylı bir analiz gerektirmektedir.

Tez çalışmasında piksel çıkarılmış görüntülerin tespiti için farklı bir yöntem geliştirilmiştir. Önerilen şerit tabanlı tespit yöntemi, orijinal ve piksel yolu çıkarılmış görüntüleri başarılı bir şekilde ayırabildiğinden adli analiz (forensic analysis) için kullanılabilir. Yerel ikili örüntüler (YİÖ), piksel yolu çıkarmanın tespitinde etkilidir. Ayrıca, enerji haritasının karekökünü alma veri aralığını daraltarak, çıkarılan özellikleri belirginleştirmektedir. Yöntemdeki temel başarı görüntülerin şeritlere ayrılarak incelenmesinden gelmektedir. Bu da, görüntülerin alt parçalarının incelenmesi tespitinde etkili olduğunu göstermektedir. Geliştirilen yöntemin başarısını özellikle de tespiti zor olan düşük oranda piksel yolu çıkarılmış görüntüler belirlemektedir. Görüntülerin şeritlere ayrılarak irdelenmesi düşük ölçekleme oranlarında da etkili sonuçlar üretmektedir. Şeritlere ayırmada gereken optimum derinlik veri tabanındaki görüntülerin boyutlarına bağlı olarak değişir. Tespit yönteminde muhtemel tüm piksel yollarını hesaba katmak yöntemi daha kararlı hale getirir. Ayrıca, sınıflandırmada kullanılacak özellik sayısı sınıflayıcının performansı üzerinde etkilidir. Özellik sayısı çok fazla ise başarımlar düşer.

Piksel yolu çıkarma yönteminin, tespiti zorlaşacak şekilde geliştirilmesi konusunda bir çalışma yapılabilir. Bu amaçla, tespitinde kullanılan özelliklerin etkileri azaltılabilir.

Ayrıca, kullanılan tespit yöntemi, kullanıcının parametre belirlemesinden bağımsız, yalnızca girdi görüntülerinin verildiği bir şekilde geliştirilebilir. Bunun için de sezgisel yöntemlerin kullanılması muhtemel çalışma konularındandır. Ek olarak, önerilen yöntem tamamen GPU tabanlı olarak geliştirilerek, özellik çıkarımı, görüntülerin piksel yolu çıkarılarak ön işlenmesi, destek vektör makinesinin eğitilmesi ve testi boyunca harcanan sürelerin en aza indirilmesi sağlanabilir. Ayrıca, piksel yolu çıkarılmış görüntülerin orijinallerinin elde edilmesini sağlayacak bir yöntem de geliştirilebilir.

## KAYNAKLAR

- [1] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, p. 10, 2007.
- [2] K. Li, B. Yan, J. Li, and A. Majumder, "Seam carving based aesthetics enhancement for photos," *Signal Process. Image Commun.*, vol. 39, pp. 509–516, 2015.
- [3] T. Cook, R. Sutton, and K. Buckley, "Automated flexion crease identification using internal image seams," *Pattern Recognit.*, vol. 43, no. 3, pp. 630–635, 2010.
- [4] R. Saabni, A. Asi, and J. El-Sana, "Text line extraction for historical document images," *Pattern Recognit. Lett.*, vol. 35, no. 1, pp. 23–33, 2014.
- [5] S. Cho, H. Choi, Y. Matsushita, and S. Lee, "Image retargeting using importance diffusion," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 977–980.
- [6] M. Rubinstein, A. Shamir, and S. Avidan, "Multi-operator Media Retargeting."
- [7] S. Luo, J. Zhang, Q. Zhang, and X. Yuan, "Multi-operator image retargeting with automatic integration of direct and indirect seam carving," *Image Vis. Comput.*, vol. 30, no. 9, pp. 655–667, 2012.
- [8] M. Xu, Z. Wang, W. Chen, and I. Shen, "ICC Image Diffusion Based Seam Carving," *J. Comput. Inf. Syst.*, vol. 8, no. 8, pp. 3285–3292, 2012.
- [9] Z. Wang, L. Wang, Y. Peng, and I. Shen, "Edge-preserving based adaptive ICC method for image diffusion," in *2010 3rd International Congress on Image and Signal Processing*, 2010, pp. 1638–1641.
- [10] Z. Karapinar Senturk and D. Akgun, "A Performance Analysis of Seam Carving Algorithm Based on Energy Function," in *Signal Processing and Communication Application Conference (SIU)*, 2016.
- [11] Y. Chen, Y. Pan, M. Song, and M. Wang, "Image retargeting with a 3D saliency model," *Signal Processing*, vol. 112, pp. 53–63, 2015.

- [12] Q. Wang and X. Li, "Shrink image by feature matrix decomposition," *Neurocomputing*, vol. 140, pp. 162–171, 2014.
- [13] W. Wang, D. Zhai, T. Li, and X. Feng, "Salient edge and region aware image retargeting," *Signal Process. Image Commun.*, vol. 29, no. 10, pp. 1223–1231, 2014.
- [14] B. Saha, T. Dasgupta, and S. Bhattacharya, "An improved content aware image resizing algorithm based on a novel adaptive seam detection technique," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 2311–2316.
- [15] P. Getreuer, "Chan-Vese Segmentation," *Image Process. Line*, vol. 2, pp. 214–224, Aug. 2012.
- [16] Q. Wang and Y. Yuan, "Learning to resize image," *Neurocomputing*, vol. 131, pp. 357–367, 2014.
- [17] D. D. Conger, M. Kumar, and H. Radha, "Multi-seam carving via seamlets," 2011, p. 78700H.
- [18] M. Kumar, D. D. Conger, R. L. Miller, J. Luo, H. Radha, M. Kumar, R. L. Miller, J. Luo, D. D. Conger, and H. Radha, "A Distortion-Sensitive Seam Carving Algorithm for Content-Aware Image Resizing," *J Sign Process Syst*, vol. 65, pp. 159–169, 2011.
- [19] X. Lin, B. Sheng, L. Ma, Y. Shen, and Z. Chen, "Seamlet carving for shape-aware image resizing," *Sci. China Inf. Sci.*, vol. 55, no. 5, pp. 1073–1081, May 2012.
- [20] T. Basha, Y. Moses, and S. Avidan, "Geometrically consistent stereo seam carving," in *2011 International Conference on Computer Vision*, 2011, pp. 1816–1823.
- [21] J.-W. Han, K.-S. Choi, T.-S. Wang, S.-H. Cheon, and S.-J. Ko, "Wavelet based seam carving for content-aware image resizing," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 345–348.
- [22] J. V. C. I. R, "Optimal bi-directional seam carving for compressibility-aware image," *J. Vis. Commun. Image Represent.*, 2016.
- [23] Z. Karapinar Senturk and D. Akgun, "A Performance Analysis for Seam Carving Algorithm," *Int. J. Adv. Stud. Comput. Sci. Eng.*, vol. 3, no. 12, pp. 5–11, 2014.

- [24] H. Hsin, T. Sung, and C. Su, "A Fast Wavelet-Based Seam Carving Algorithm for Image Resizing," vol. 2, no. 5, pp. 119–123, 2014.
- [25] I. Kim, J. Zhai, Y. Li, and W. Chen, "Optimizing seam carving on multi-GPU systems for real-time content-aware image resizing," *J. Supercomput.*, vol. 71, no. 9, pp. 3500–3524, 2015.
- [26] A. Sarkar, L. Nataraj, and B. S. Manjunath, "Detection of Seam Carving and Localization of Seam Insertions in Digital Images."
- [27] C. Fillion and G. Sharma, "Detecting content adaptive scaling of images for forensic applications," *IS&T/SPIE Electron.*, 2010.
- [28] W. Lu and M. Wu, "Seam carving estimation using forensic hash," in *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security - MM&Sec '11*, 2011, p. 9.
- [29] S. J. Ryu, H. Y. Lee, and H. K. Lee, "Detection of Content-Aware Image Resizing Using Seam Properties," *Appl. Mech. Mater.*, vol. 284–287, pp. 3074–3078, Jan. 2013.
- [30] M. S. Gerald Schaefer, "UCID - An Uncompressed Colour Image Database," *In Storage and Retrieval Methods and Applications for Multimedia 2004, volume 5307 of Proceedings of SPIE.* .
- [31] J.-D. Wei, Y.-J. Lin, Y.-J. Wu, and L.-W. Kang, "A patch analysis approach for seam-carved image detection," in *ACM SIGGRAPH 2013 Posters on - SIGGRAPH '13*, 2013, p. 1.
- [32] S.-J. RYU, H.-Y. LEE, and H.-K. LEE, "Detecting Trace of Seam Carving for Forensic Analysis," *IEICE Trans. Inf. Syst.*, vol. E97.D, no. 5, pp. 1304–1311, 2014.
- [33] J.-D. Wei, Y.-J. Lin, and Y.-J. Wu, "A patch analysis method to detect seam carved images," 2014.
- [34] P.-Y. Jiang and H.-J. C. and J.-D. Wei, "Using Patch Analysis Methods to Detect Images Tampered with Seam Insertion." The Society of Digital Information and Wireless Communication, pp. 187–190, 2015.
- [35] W.-L. Chang, T. K. Shih, and H.-H. Hsu, "Detection of seam carving in JPEG images," in *2013 International Joint Conference on Awareness Science and Technology & Ubi-Media Computing (iCAST 2013 & UMEDIA 2013)*, 2013, pp. 632–638.



- [36] W. Fei, Y. Gaobo, L. Leida, X. Ming, and Z. Dengyong, "Detection of seam carving-based video retargeting using forensics hash," *Secur. Commun. Networks*, vol. 8, no. 12, pp. 2102–2113, Aug. 2015.
- [37] L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF," *Int. J. Image Process.*, vol. 3, no. 4, pp. 143–152, 2009.
- [38] K. Wattanachote, T. K. Shih, W. L. Chang, and H. H. Chang, "Tamper Detection of JPEG Image Due to Seam Modifications," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 12, pp. 2477–2491, 2015.
- [39] T. Yin, G. Yang, L. Li, D. Zhang, and X. Sun, "Detecting seam carving based image resizing using local binary patterns," *Comput. Secur.*, vol. 55, pp. 130–141, 2015.
- [40] Z. Karapinar Senturk and D. Akgun, "Yerel İkili Örüntüler Tabanlı Seam Carving Tespit Yaklaşımının Yatay Ve İki Yönlü Seam Carving Açısından İncelenmesi," in *1st International Conference on Engineering Technology and Applied Sciences*, 2016.
- [41] Z. Karapinar Senturk and D. Akgun, "Seam Carving Tespitinde Görüntü Boyutunun Karşılaştırmalı Analizi," in *ASYU 2016 (Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu)*, 2016.
- [42] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [43] M. S. Chris Harris, "A combined corner and edge detector," *In Proc. of Fourth Alvey Vision Conference*.
- [44] D. DeCarlo, A. Santella, D. DeCarlo, and A. Santella, "Stylization and abstraction of photographs," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques - SIGGRAPH '02*, 2002, vol. 21, no. 3, p. 769.
- [45] Y. Chen, Y. Pan, M. Song, and M. Wang, "Improved seam carving combining with 3D saliency for image retargeting," *Neurocomputing*, vol. 151, pp. 645–653, 2014.
- [46] Q. Wang, Y. Yuan, and L. Wang, "High quality image resizing," *Neurocomputing*, vol. 131, pp. 348–356, 2014.
- [47] E. Dekkers and L. Kobbelt, "Geometry seam carving," *Comput. Des.*, vol. 46, pp. 120–128, 2014.

- [48] Y. Chen, Y. Pan, M. Song, and M. Wang, "Image retargeting with a 3D saliency model," *Signal Processing*, vol. 112, pp. 53–63, 2015.
- [49] M. Pietikäinen, "Local Binary Patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.
- [50] S. Abbasi and F. Tajeripour, "Detection of brain tumor in 3D MRI images using local binary patterns and histogram orientation gradient," *Neurocomputing*, no. October 2015, pp. 0–1, 2016.
- [51] S. V. da Rocha, G. Braz Junior, A. C. Silva, A. C. de Paiva, and M. Gattass, "Texture analysis of masses malignant in mammograms images using a combined approach of diversity index and local binary patterns distribution," *Expert Syst. Appl.*, vol. 66, pp. 7–19, 2016.
- [52] B. Patel, R. P. Maheshwari, and R. Balasubramanian, "Multi-quantized local binary patterns for facial gender classification," *Comput. Electr. Eng.*, vol. 54, pp. 271–284, 2015.
- [53] A. Hadid, J. Ylioinas, M. Bengherabi, M. Ghahramani, and A. Taleb-Ahmed, "Gender and texture classification: A comparative analysis using 13 variants of local binary patterns," *Pattern Recognit. Lett.*, vol. 68, pp. 231–238, 2015.
- [54] L. Liu, P. Fieguth, G. Zhao, M. Pietikäinen, and D. Hu, "Extended local binary patterns for face recognition," *Inf. Sci. (Ny)*, vol. 358–359, pp. 56–72, 2016.
- [55] F. Bianconi, E. González, and A. Fernández, "Dominant local binary patterns for texture classification: Labelled or unlabelled?," *Pattern Recognit. Lett.*, vol. 65, pp. 8–14, 2015.
- [56] R. Mehta and K. Egiazarian, "Dominant Rotated Local Binary Patterns (DRLBP) for texture classification," *Pattern Recognit. Lett.*, vol. 71, pp. 16–22, 2016.
- [57] Y. Zhang and C. Hua, "Driver fatigue recognition based on facial expression analysis using local binary patterns," *Opt. - Int. J. Light Electron Opt.*, vol. 126, no. 23, pp. 4501–4505, 2015.
- [58] I. Choi and D. Kim, "A variety of local structure patterns and their hybridization for accurate eye detection," *Pattern Recognit.*, vol. 61, pp. 417–432, 2017.

- [59] Y. Kaya, L. Kayci, and M. Uyar, “Automatic identification of butterfly species based on local binary patterns and artificial neural network,” *Appl. Soft Comput.*, vol. 28, no. 2015, pp. 132–137, 2015.
- [60] Z. Wen, Z. Li, Y. Peng, and S. Ying, “Virus image classification using multi-scale completed local binary pattern features extracted from filtered images by multi-scale principal component analysis,” *Pattern Recognit. Lett.*, vol. 79, pp. 25–30, 2016.
- [61] V. V. Nabiyev and G. Asuman, “LBP Yardımıyla Görüntüdeki Kişinin Yaşının Bulunması,” *Çankaya Univ. J. Sci. Eng.*, vol. 8, no. 1, pp. 27–41, 2011.
- [62] V. Nabiyev and F. Koçak, “YEREL İKİLİ ÖRNEKLERLE YÜZ GÜZELLİĞİ KARARI,” in *Fırat Elektrik-Elektronik Bilgisayar Sempozyumu*, 2011, pp. 94–98.
- [63] V. Kecman, “Support Vector Machines – An Introduction,” 2005, pp. 1–47.
- [64] R. Berwick, “Berwick2003 - An Idiot’s guide to Support vector machines (SVMs) R. Berwick,” 2003. [Online]. Available: <https://www.coursehero.com/file/10996787/Berwick2003/>. [Erişim Tarihi: 12.11.2016].
- [65] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [66] T. Fletcher, “Support Vector Machines Explained,” 2008.
- [67] J. P. Harvey, “GPU acceleration of object classification algorithms using NVIDIA CUDA,” 2009.
- [68] J. Owens, U. Davis, and D. Luebke, “Parallel Programming and Computing Platform|CUDA|NVIDIA.” [Online]. Erişim Tarihi:15.10.2016 [www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html).
- [69] PricenFees, “NVIDIA CUDA.” [Online]. Erişim Tarihi:12.10.2016 <https://www.pricenfees.com/digit-life-archives/nvidia-cuda>.
- [70] J. Hennessy, D. Patterson, and K. Asanović, *Computer architecture: a quantitative approach*, 5th ed. Morgan Kaufmann Publishers Inc., 2012.
- [71] NVIDIA, “GPU vs CPU? What is GPU Computing?” [Online]. Erişim Tarihi:14.06.2016 [www.nvidia.com/object/what-is-gpu-computing.html](http://www.nvidia.com/object/what-is-gpu-computing.html).

- [72] A. Carpenter, "CUSVM: A CUDA IMPLEMENTATION OF SUPPORT VECTOR CLASSIFICATION AND REGRESSION."
- [73] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Int. Jt. Conf. Artif. Intell.*, vol. 14, no. 12, pp. 1137–1143, 1995.
- [74] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit.*, vol. 27, pp. 861–874, 2006.
- [75] J. P. Egan, *Signal Detection Theory and ROC-analysis*. Academic Press, 1975.
- [76] J. Swets, R. Dawes, and J. Monahan, "Better Decisions through Science," *Sci. Am.*, vol. 283, no. 4, pp. 82–87, 2000.
- [77] J. Swets, "Measuring the accuracy of diagnostic systems," *Science (80-. )*, vol. 240, no. 4857, pp. 1285–1293, 1988.
- [78] K. H. Zou, "Receiver Operating Characteristic (ROC) Literature Research," 2002. [Online]. Erişim Tarihi:12.12.2015  
<https://www.spl.harvard.edu/archive/spl-pre2007/pages/ppl/zou/roc.html>.
- [79] M. M. Siddiqui, G. Srivastava, and S. H. Saeed, "Diagnosis of insomnia sleep disorder using short time frequency analysis of PSD approach applied on EEG signal using channel ROC-LOC," *Sleep Sci.*, 2016.
- [80] A. D. Althouse, "Statistical graphics in action: Making better sense of the ROC curve," *Int. J. Cardiol.*, vol. 215, pp. 9–10, 2016.
- [81] J. Yin, J. Xie, X. Guo, L. Ju, Y. Li, and Y. Zhang, "Plasma metabolic profiling analysis of cyclophosphamide-induced cardiotoxicity using metabolomics coupled with UPLC/Q-TOF-MS and ROC curve," *J. Chromatogr. B*, vol. 1033–1034, pp. 428–435, 2016.
- [82] H. Shang, Y. Jiang, F. Li, H. MacMahon, and J. Wang, "ROC Curve for Extremely Subtle Lung Nodules on Chest Radiographs Confirmed by CT Scan," *Acad. Radiol.*, vol. 23, no. 3, pp. 297–303, 2016.
- [83] K. A. Spackman, "Signal detection theory: valuable tools for evaluating inductive learning," in *Proceedings of the sixth international workshop on Machine learning*, 1989, pp. 160–163.

- [84] D. E. Marasco and C. E. Kontokosta, “Applications of machine learning methods to identifying and predicting building retrofit opportunities,” *Energy Build.*, vol. 128, pp. 431–441, 2016.
- [85] A. B. Rosenkrantz, A. M. Doshi, L. A. Ginocchio, and Y. Aphinyanaphongs, “Use of a Machine-learning Method for Predicting Highly Cited Articles Within General Radiology Journals,” *Acad. Radiol.*, vol. 23, no. 12, pp. 1573–1581, 2016.
- [86] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015.
- [87] C. Kotsavasiloglou, N. Kostikis, D. Hristu-Varsakelis, and M. Arnaoutoglou, “Machine learning-based classification of simple drawing movements in Parkinson’s disease,” *Biomed. Signal Process. Control*, vol. 31, pp. 174–180, 2017.
- [88] Z. Karapinar Senturk and D. Akgun, “Seam Carving Based Image Resizing Detection Using Hybrid Features,” *Tech. Gaz.*, 2017.
- [89] J. Eaton, D. Bateman, and S. Hauberg, *Gnu octave*. London: Network theory, 1997.
- [90] C.-C. Chang and C.-J. Lin, “LIBSVM: A Library for Support Vector Machines.”

## ÖZGEÇMİŞ

Zehra Karapınar Şentürk, 24.01.1987'de Ankara'da doğdu. İlk, orta, lise ve üniversite eğitimini Ankara'da tamamladı. 2004 yılında Mimar Sinan Lisesi'nden mezun oldu. 2004 yılında başladığı Atılım Üniversitesi Bilgisayar Mühendisliği Bölümü'nü 2009'da Matematik yan dalı ile birlikte tamamladı. 2010 yılında araştırma görevlisi olarak Düzce Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde göreve başladı. Aynı yıl Düzce Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümü'nde yüksek lisans eğitimine başladı ve 2011 sonunda mezun oldu. 2012 yılında Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği doktora programına kabul edilerek doktora eğitimine başladı. Halen Düzce Üniversitesi'nde araştırma görevlisi olarak görev yapmaktadır. Evli ve bir çocuk annesidir.