

**T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SÜPERPİKSEL ALGORİTMALARININ GÜRÜLTÜ DUYARLILIĞI VE
NESNE BÖLÜTLEME PERFORMANSININ İNCELENMESİ**

Fadime ÖZER

**Danışman
Yrd. Doç. Dr. Ufuk ÖZKAYA**

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
ISPARTA-2018**



© 2018 [Fadime ÖZER]

TEZ ONAYI

Fadime ÖZER tarafından hazırlanan " **Süperpiksel Algoritmalarının Gürültü Duyarlılığı ve Nesne Bölütleme Performansının İncelenmesi** " adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak başarı ile savunulmuştur.

Danışman

Yrd. Doç. Dr. Ufuk ÖZKAYA
Süleyman Demirel Üniversitesi

Jüri Üyesi

Doç. Dr. Özlem COŞKUN
Süleyman Demirel Üniversitesi

Jüri Üyesi

Yrd. Doç. Dr. Övünç POLAT
Akdeniz Üniversitesi

Enstitü Müdürü

Prof. Dr. Yasin TUNCER

TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Fadime ÖZER



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET	ii
ABSTRACT	iv
TEŞEKKÜR.....	vi
ŞEKİLLER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	xi
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	5
2.1. Su Seti Tabanlı Algoritmalar	5
2.2. Yoğunluk Tabanlı Algoritmalar	7
2.3. Graf Tabanlı Algoritmalar	9
2.4. Kontur Tabanlı Algoritmalar	14
2.5. Yol Tabanlı Algoritmalar	16
2.6. Kümeleme Tabanlı Algoritmalar	18
2.7. Enerji Tabanlı Algoritmalar	22
2.8. Diğer Çalışmalar	24
3. YAPILAN ÇALIŞMALAR.....	30
3.1. Gürültü	32
3.1.1. Gauss Gürültüsü	33
3.1.2. Tuz&Biber Gürültüsü	35
3.1.3. Bulanıklaştırma	36
3.2. Filtre.....	41
3.2.1. Medyan Filtre	42
3.2.2. Wiener Filtre	44
3.3. Süperpiksel Algoritmaları	46
3.3.1. Basit Doğrusal Yinelemeli Kümeleme Algoritması (SLIC)	47
3.3.2. Doğrusal Spektral Kümeleme Algoritması (LSC)	50
3.3.3. Düzenliliği Korunmuş Süperpiksel Algoritması (RPS).....	53
3.3.4. Entropi Oranı Kümeleme Algoritması (ERC)	55
3.3.5. Turbo Piksel Algoritması (TP)	57
3.4. Test İmgeleri	59
4. BAŞARI METRİKLERİ	61
4.1. Kesin Referans	61
4.2. Sınır Örtüşmesi	62
4.3. Eksik Bölütleme Hatası.....	63
4.4. Erişilebilir Bölütleme Doğruluğu	64
4.5. Açıklık Değişimi	65
4.6. Doluluk Derecesi(Derlitopluluk)	66
4.7. Boyut Değişimi.....	66
4.8. Küme İçi Değişimi	67
4.9. PSNR.....	67
5. SONUÇLAR.....	68
6. TARTIŞMA.....	94
7. KAYNAKLAR	96
ÖZGEÇMİŞ.....	100

ÖZET

Yüksek Lisans Tezi

SÜPERPİKSEL ALGORİTMALARININ GÜRÜLTÜ DUYARLILIĞI VE NESNE BÖLÜTLEME PERFORMANSININ İNCELENMESİ

Fadime ÖZER

Süleyman Demirel Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Ufuk ÖZKAYA

Süperpikseller, özellikle imge ve video bölütleme uygulamalarında kullanılan; bir imgeye veya videoya ait benzer renk, yoğunluk ve doku özellikleri gösteren piksellerden oluşan görsel yapılardır. Süperpikseller özellikle biyomedikal, askeri ve uzaktan algılama çalışmalarında bölütleme, nesnelere tespiti, takibi, arka planların çıkarılması, belirginlik tespiti vb. uygulamalarda kullanılmaktadır. Literatürdeki süperpiksel yöntemlerinin bölütleme performansları ideal ve gürültü içermeyen imgelere uygulanarak elde edilen sonuçlara göre ölçülmekte ve sunulmaktadır. Ancak, pratik uygulamalarda hem iletim ortamı hem de görüntüleme sistemlerindeki lens hücre hatalarından dolayı imgeler çoğunlukla gürültü içermektedirler. Dolayısıyla önerilen bir süperpiksel algoritmasının sadece ideal için değil aynı zamanda gürültü içeren imgeler için de bölütleme performanslarının analiz edilmesi gerekmektedir.

Bu tez çalışmasında literatürde en yaygın şekilde kullanılan süperpiksel algoritmalarının gürültülü imgeler için bölütleme performansı ve gürültü bağışıklığı incelenmiştir. Ayrıca, gürültülü imgeler için bölütleme uygulamalarında ön işlem elemanı olarak filtre kullanımının bölütleme performansına olan etkisi de incelenmiştir. Bu amaçla, farklı yaklaşımlarla geliştirilmiş beş süperpiksel algoritması imge bölütleme veri setinden seçilen test imgelerine uygulanmıştır. Test imgesi olarak farklı doku, renk ve histogram davranışlarına sahip on beş imge seçilmiştir. Gerçekleştirilen uygulamalar sonucunda, farklı gürültü tip ve seviyelerinin süperpiksel algoritmalarının bölütleme performanslarına etkisi ve süperpiksel algoritmalarının gürültü bağışıklığı incelenmiştir. Ayrıca, ön işlem elemanı olarak filtre kullanımının süperpiksel algoritmalarının bölütleme performansına olan etkisi elde edilen başarı metrikleri ile ortaya konmuştur.

Elde edilen sonuçlar ışığında gürültülü imgelerde nesnelere ayıran sınır bilgisinin belirginliğinin azalması nedeniyle süperpiksel algoritmalarının bölütleme performansının önemli derecede azaldığı görülmüştür. İncelenen süperpiksel algoritmaları arasında bölütleme performansı ve gürültü bağışıklığı bağlamında gerekli analiz ve kıyaslamalar gerçekleştirilmiştir. Ön işlem elemanı olarak filtre

kullanımının gürültü ve filtre tipine baęlı olarak bölütleme performansını arttırdığı gözlemlenmiştir.

Anahtar Kelimeler: Süperpiksel, imge bölütleme, gürültü.

2018, 100 sayfa



ABSTRACT

M.Sc. Thesis

INVESTIGATION OF NOISE SENSITIVITY AND OBJECT SEGMENTATION PERFORMANCES OF SUPERPIXEL ALGORITHMS

Fadime ÖZER

**Süleyman Demirel University
Graduate School of Natural and Applied Sciences
Department of Electronics and Communications Engineering**

Supervisor: Asst. Prof. Dr. Ufuk ÖZKAYA

Superpixels, used especially in image and video segmentation applications, are the visual structures composed of pixels having same color, intensity and texture behavior. Superpixels are used especially in biomedical, military and remote sensing studies, applications such as segmentation, detection and tracking of object, background subtraction, saliency detection etc. The segmentation performance of superpixel methods in the literature are measured and presented by applying to ideal and noise-free images. However, in practical applications, images often contain noise due to both noise components in the transmission environment and the lens cell errors in the imaging systems. Therefore, the segmentation performances of a proposed superpixel algorithm should be analysed not only for ideal images, but also for noise-containing images.

In this thesis study, segmentation performance for noisy images and noise sensitivity of the most common used superpixel algorithms in the literature are examined. Moreover, in segmentation applications for noisy images, the effect of filter usage as a pre-processing element on segmentation performance is examined. For this purpose, five superpixel algorithms developed with different approaches have been applied to selected test images from the image segmentation data set. Fifteen images that have different texture, colour and histogram characteristics are selected as test images. As a result of the applications performed, the effects of different noise types and levels on the segmentation performances of the superpixel algorithms and noise sensitivity of the superpixel algorithms are examined. Furthermore, the effect of filter usage as a pre-processing element on the segmentation performance of superpixel algorithms is demonstrated by the achieved success metrics.

Obtained results showed that the segmentation performance of superpixel algorithms decreased significantly due to the decreasing of significance of boundary information that separates objects in noisy images. Among the superpixel algorithms examined, the necessary analyses and comparisons are made in terms of segmentation performance and noise sensitivity. It has been

observed that the usage of filters as pre-processing element increases the segmentation performance depending on the noise and filter type.

Keywords: Superpixel, image segmentation, noise.

2018, 100 pages



TEŐEKKÖR

Bu arařtırma için beni yönlendiren, karşılařtıđım zorlukları bilgi ve tecrübesi ile ařmamda yardımcı olan deđerli Danıřman Hocam Yrd. Doç. Dr. Ufuk ÖZKAYA' ya sonsuz teőekkürlerimi sunarım.

4692-YL2-16 No'lu Proje ile tezimi maddi olarak destekleyen Süleyman Demirel Üniversitesi Bilimsel Arařtırma Projeleri Yönetim Birimi Başkanlıđı'na teőekkür ederim.

Tezimin her ařamasında beni yalnız bırakmayan ve her zaman destekleyen annem Faden ÖZER'e, babam Ahmet ÖZER'e, ablam Zeliha İNAN'a, abim Ufuk İNAN'a, biricik yeđerim Efe İNAN'a ve canım Hidayet ÖZEREN'e sonsuz sevgi ve saygılarımı sunarım. Çalışmalarım boyunca çalışmamı en çok anlattıđım ve yardımlarını esirgemeyen OPTİMA Arařtırma Grubu'ndan arkadaşlarım Gizem DURSUN ve Ali Alper DEMİR'e çok teőekkür ederim.

Fadime ÖZER
ISPARTA, 2018

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. MR imgesi ve bölütlenmiş MR imgesi.....	25
Şekil 2.2. Farklı prostat imgeleri ve bölütlenmiş prostat imgeleri.....	27
Şekil 2.3. (a) Sentetik veri, (b) Süperpiksellere ayrılmış sentetik veri, (c) SLIC ve NFINDR yöntemi ile gürültü giderimi, (d) SLIC ve VCA yöntemi ile gürültü giderimi	28
Şekil 2.4. (a) Hyperion veri kümesi, (b) Süperpiksellere ayrılmış gerçek veri, (c) SLIC ve NFINDR yöntemi ile gürültü giderimi, (d) SLIC ve VCA yöntemi ile gürültü giderimi	28
Şekil 3.1. Orijinal imgeler için bölütleme blok diyagramı.....	31
Şekil 3.2. Gürültülü imgeler için bölütleme blok diyagramı	31
Şekil 3.3. Filtreli imgeler için bölütleme blok diyagramı	32
Şekil 3.4. Sentetik orijinal imge ve gauss gürültülü imge ve histogram dağılımları.....	34
Şekil 3.5. Test imgesi için orijinal imge ve gauss gürültülü imge ve histogram dağılımları.....	34
Şekil 3.6. Sentetik orijinal imge ve tuz&biber gürültülü imge ve histogram dağılımları.....	36
Şekil 3.7. Test imgesi için orijinal imge ve tuz&biber gürültülü imge ve histogram dağılımları.....	36
Şekil 3.8. Sentetik orijinal imge ve bulanıklaştırılmış imge ve histogram dağılımları.....	37
Şekil 3.9. Test imgesi için orijinal imge ve bulanıklaştırılmış imge ve histogram dağılımları.....	37
Şekil 3.10. Sentetik orijinal imge, gauss gürültülü imge ve medyan filtreli imge ve histogram dağılımları	42
Şekil 3.11. Sentetik orijinal imge, tuz&biber gürültülü imge ve medyan filtreli imge ve histogram dağılımları	43
Şekil 3.12. Sentetik orijinal imge, bulanıklaştırılmış imge ve medyan filtreli imge ve histogram dağılımları	44
Şekil 3.13. Sentetik orijinal imge, gauss gürültülü imge ve wiener filtreli imge ve histogram dağılımları	45
Şekil 3.14. Sentetik orijinal imge, tuz&biber gürültülü imge ve wiener filtreli imge ve histogram dağılımları	46
Şekil 3.15. Sentetik orijinal imge, bulanıklaştırılmış imge ve wiener filtreli imge ve histogram dağılımları	46
Şekil 3.16. SLIC algoritmasının m değerine göre 200 süperpiksel için sonuçları	49
Şekil 3.17. SLIC algoritmasının farklı süperpiksel sayıları için süperpiksel performansları	50
Şekil 3.18. LSC algoritmasının farklı süperpiksel sayıları için süperpiksel performansları	53
Şekil 3.19. RPS algoritmasının farklı süperpiksel sayıları için süperpiksel performansları	55
Şekil 3.20. ERC algoritmasının farklı süperpiksel sayıları için süperpiksel performansları	57

Şekil 3.21. TP algoritmasının farklı süperpiksel sayıları için süperpiksel performansları	59
Şekil 3.22. Tez çalışmasında kullanılan test imgeleri	60
Şekil 4.1. Test imgeleri ve kesin referans gösterimi-1	61
Şekil 4.2. Test imgeleri ve kesin referans gösterimi-2	62
Şekil 4.3. Test imgeleri ve kesin referans gösterimi-3	62
Şekil 4.4. Sınır örtüşmesi için sentetik gösterim	63
Şekil 4.5. Eksik bölütleme hatası için sentetik gösterim	64
Şekil 4.6. Erişilebilir bölütleme doğruluğu için sentetik gösterim	65
Şekil 4.7. Açıklık değişimi için sentetik gösterim	66
Şekil 5.1. Tez çalışmasının genel blok diyagramı	68
Şekil 5.2. Orijinal imgeler için süperpiksel sayısına bağlı olarak algoritmaların performanslarının değerlendirilmesi	69
Şekil 5.3. Süperpiksel algoritmalarının 100 süperpiksel için gürültüye bağlı olarak süperpiksel performanslarının incelenmesi (a) Orijinal imge, (b) Gauss gürültülü imge, (c) Tuz&biber gürültülü imge, (d) Bulanıklaştırılmış imge.....	70
Şekil 5.4. SLIC algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi	71
Şekil 5.5. SLIC algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	72
Şekil 5.6. SLIC algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	73
Şekil 5.7. LSC algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	74
Şekil 5.8. LSC algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	75
Şekil 5.9. LSC algoritmasının bulanıklaştırma etkisi bağlı olarak metrik değerlerinin incelenmesi	76
Şekil 5.10. ERC algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	77
Şekil 5.11. ERC algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	78
Şekil 5.12. ERC algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi	79
Şekil 5.13. RPS algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	80
Şekil 5.14. RPS algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	81
Şekil 5.15. RPS algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi	82
Şekil 5.16. TP algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi.....	83
Şekil 5.17. TP algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi	84
Şekil 5.18. TP algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi	85
Şekil 5.19. Süperpiksel algoritmalarının gauss gürültüsüne bağlı değişimi.....	86
Şekil 5.20. Süperpiksel algoritmalarının tuz&biber gürültüsüne bağlı değişimi	87

Şekil 5.21. Süperpiksel algoritmalarının bulanıklaştırma etkisine bağlı değişimi.....	88
Şekil 5.22. Beyin MR görüntüsünün süperpiksel algoritmaları için sonuçları-1	89
Şekil 5.23. Beyin MR görüntüsünün süperpiksel algoritmaları için sonuçları-2	89
Şekil 5.24. Gürültülü imgelerin filtreler ile iyileştirme performansının incelenmesi.....	90



ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1. İncelenen süperpiksel algoritmalarının listesi.....	29
Çizelge 5.1. Gürültülü imgelere uygulanan filtrelerin ortalama PSNR değeri.....	90
Çizelge 5.2. Filtreli imgelerin SÖ başarı metriğinin incelenmesi	91
Çizelge 5.3. Filtreli imgelerin EBD başarı metriğinin incelenmesi.....	91
Çizelge 5.4. Filtreli imgelerin AD başarı metriğinin incelenmesi.....	92
Çizelge 5.5. Filtreli imgelerin EBH başarı metriğinin incelenmesi.....	93



SİMGELER VE KISALTMALAR DİZİNİ

AD	Açıklık değişimi
BD	Boyut değişimi
BSD	Berkeley veri kümesi
CIE	Uluslararası aydınlatma komisyonu
CT	Bilgisayarlı tomografi
DD	Doluluk derecesi
EBD	Erişilebilir bölütleme doğruluğu
EBH	Eksik bölütleme hatası
ERC	Entropi oranı kümeleme algoritması
FCM	Bulanık kümeleme
KİD	Küme içi değişimi
KOU	Kenara olan ortalama uzaklık
LRW	Tembel rasgele yürüyüş
LSC	Doğrusal spektral kümeleme algoritması
MR	Manyetik rezonans
MRI	Manyetik rezonans görüntüleme
MST	En küçük yayılan ağaç
NCUT	Normalize kesme algoritması
PDF	Olasılık yoğunluk fonksiyonu
PSNR	Tepe sinyal gürültü oranı
RPS	Düzenliliği korunmuş süperpiksel algoritması
RW	Rasgele yürüyüş
SLIC	Basit doğrusal yinelemeli kümeleme algoritması
SÖ	Sınır örtüşmesi
TP	Turbo piksel algoritması
VCA	Zirve bileşen analizi

1. GİRİŞ

İmge bölütleme, imgenin anlamlı bölütlere ayrılmasıyla elde edilen görüntü işleme ve bilgisayarlı görü sistemlerinde yaygın olarak kullanılan bir yöntemdir. Elde edilen bölütler ile kenar tespiti, nesne tespiti, arka plan çıkarımı, hareket analizi gibi uygulamalarda kullanılarak imge daha anlamlı hale getirilmektedir. Görüntü işleme tekniklerinde imge bölütleme genellikle eşik değeri, piksel gruplarının genişlemesi veya kenarlarının bulunması için filtreler kullanılarak bölütler oluşturulmaktadır. Örneğin gri seviyeli bir imge için iki bölüte ayrılabilmesi için bir eşik değeri belirlenerek eşik değerinden küçük yoğunluk değerine sahip piksellere bir etiket, büyük değerlere ise bir başka etiket verilerek iki bölüt oluşturulmaktadır. Piksel gruplarının genişlemesiyle oluşturulan bölütlerde ise imgenin dokusal özelliklerinden yararlanılmaktadır. Bu yöntemler kullanılarak oluşturulan bölütler farklı boyutlarda, düzensiz şekillerde oluşmaktadır. Bir ön işlem olarak kullanıldığında imge bölütleme sonucunda oluşan bölütler ile işlem yapılırken ve tanımlarken aynı boyutlarda olması istenilmektedir. Bunun nedenlerinden biri bölüt tanımlanırken daha veri kullanılarak tanımlanmak istenmektedir. İmge bölütleme işlemi sonucunda oluşan bölütlerin düzensiz ve farklı boylarda olması probleminin üstesinden gelmek için bölütlemenin bir alt çalışması olan süperpikseller geliştirilmiştir. Süperpikseller, özellikle imge ve video bölütleme uygulamalarında kullanılan; bir imgeye ait renk, yoğunluk ve doku özellikleri benzerlik gösteren ve yaklaşık aynı boyutlarda piksellerden oluşan görsel yapılardır. Süperpikselin sahip olduğu bu özellikler ile bir imgenin veya videonun daha az veriyle ifade edilebilmesini mümkün kılmaktadır. İmge işleme uygulamalarında süperpiksel kullanılmasının başlıca amacı işlem yükünü ve işlem süresini azaltmasıdır. Örneğin bir imgeyi işlerken imgedeki her bir piksel ile işlem yapmak oldukça maliyetlidir. Fakat bir imgedeki binlerce pikselden yüzlerce süperpiksele dönüştürülerek işlem yükünün ve süresinin ne kadar düştüğünün fark edilebilir şekilde gözlenmesi mümkündür. Bu yüzden süperpiksel imge işleme uygulamalarında oldukça kullanılmaktadır. Bu uygulamalar nesne tanıma [Li vd. (2012)], bölütleme [Zhang vd. (2014)], görsel izleme [Yang vd. (2014)], vücut modeli tahmini [Mori (2005)],

kenar tespiti [Sun vd. (2015), Levinshtein vd. (2010)], gürültü temizlenmesi olarak örnek verilebilir.

Bilgisayarlı görü sistemlerinde, sinyal işleme, imge ve video işleme alanında veri toplama ve veri aktarımı sırasında gürültü ne yazık ki kaçınılmazdır. İmge veya ses üzerindeki istenmeyen değişimlere gürültü denilmektedir. Diğer bir deyişle gürültü imgedeki piksel değerindeki bozulmayı yani değişimini ifade etmektedir. Gürültü haberleşme sistemlerinin kaçınılmaz bir gerçeğidir. İmge üzerinde süperpiksellere bölme işlemi yapılırken hem gönderici hem de alıcı tarafta aynı süperpiksellerin oluşması istenilmektedir. Bunun nedeni süperpikseller eğer bir ön işleme tekniği olarak kullanılırsa elde edilen sonuçların aynı olması beklenmektedir. Ayrıca imge bölütlemenin en yoğun kullanıldığı alanlardan biri de biyomedikal imge bölütlemesidir. Biyomedikal imge bölütleme genellikle 2B veya 3B imgede sınırların otomatik olarak bulunması ile tanımlanmaktadır. Medikal imgelerdeki değişiklikler, insan anatomisinin farklı olması gibi nedenlerden dolayı bölütleme oldukça zorlaşmaktadır. Ayrıca elde edilen imgeler manyetik rezonans (MR), bilgisayarlı tomografi (CT), X-ışını, mikroskopi, ultrasonik cihaz gibi farklı cihazlardan elde edilen imgelerin farklı özelliklere sahip olması da medikal imgelerin bölütlenmesini zorlaştırmaktadır. Biyomedikal cihazlardan alınan imgelerin gürültü içermesi nedeniyle bölütleme işleminin başarılı bir şekilde sonuçlandırılabilmesi için ya ön işlem olarak filtrelerin kullanılması ya da gürültü bağışıklığı yüksek olan bölütleme yöntemlerinin kullanılması gerekir. Günümüzde yaygın olarak çalışılan konulardan biri de tarım, gıda, haritalama gibi birçok uygulama alanı olan hiperspektral görüntülerin işlenmesidir. Uzaktan algılama yöntemine dayanan hiperspektral görüntüler farklı dalga boylarında spektral bilgiler taşıyan ardışık yüzlerce banttan oluşmaktadır. Bantlardan oluşan bu hiperspektral görüntüler atmosferik etki veya sensörlerden kaynaklı olarak yüksek seviyede gürültü içermektedirler. Biyomedikal imgelerde olduğu gibi doğru bir sonuca ulaşmak için hiperspektral görüntüler ya filtrelerden geçirilmeli ya da gürültü bağışıklığı yüksek süperpiksel algoritmaları kullanılmalıdır.

Literatürde birçok süperpiksel algoritması bulunmaktadır. Bu algoritmalar farklı problemler için oluşturularak, problemde öncelikli özelliğe bağlı olarak oluşturulmaktadır. Algoritmalar genellikle şekilsel ve boyutsal düzenlilik, homojenlik gibi kriterler dikkate alınarak oluşturulmaktadır. Literatürdeki çalışmalar göz önünde bulundurulduğunda veri tabanlarında kullanılan orijinal imgeler üzerinde süperpiksel işlemleri gerçekleştirilmektedir. Fakat literatürde bulunan az da olsa gürültülü imgelerin süperpikselle ayrılması üzerinde çalışılmıştır. Literatürde gürültü içeren imgelerin bölütlenmesi amacıyla gerçekleştirilen çalışmalara örnek olarak gürültüye dayanıklı FCM (Fuzzy C-Means) tabanlı bir algoritma verilebilir [Depotovic vd. (2010)]. Çalışmada imgeye ait bölütler hem benzerlik ölçüsüne hem de üyelik fonksiyonuna bağlı olarak iteratif bir şekilde oluşturulmaktadır. Bir diğer çalışmada imge bölütleme problemlerinde en çok kullanılan süperpiksel algoritmalarının gürültüsüz ve gürültülü geniş bir imge havuzu için performansları incelenmiştir [Stutz vd. (2017)]. Bu çalışmada algoritmaların performansları karşılaştırılırken gauss gürültüsü, tuz&biber gürültüsü ve bulanıklığın etkileri incelenmiştir. Diğer bir çalışmada ise mikroskopik görüntülerde farklı gürültü türlerinin süperpiksel algoritmalarının performansındaki değişim incelenmiştir [Roels vd. (2016)]. Süperpiksel algoritmalarının gürültülü verilerde sınır bilgisini kaybetmesi problemi gürültü filtreleme işlemi ile iyileştirilmiştir. Diğer taraftan süperpiksel algoritmalarının diğer bir kullanım alanı ise gürültü tahminidir [Wu ve Chang (2015), Fu vd. (2014)]. İmgedeki gürültü seviyesini kestirmek oldukça zor olduğundan imge daha küçük parçalara yani süperpikselle ayrılarak her bir süperpikselin gürültü tahmini yapılarak daha doğru sonuçlar elde edilmektedir.

Tez çalışmasının amacı literatürdeki süperpiksel algoritmalarından farklı olarak orijinal imgeler yerine gürültülü imgeler üzerinde süperpiksel algoritmalarının performansının incelenmesidir. Bunun için bu çalışması 3 ana bölümden oluşmaktadır. İlk çalışma orijinal imgelerin süperpiksel performanslarının incelenmesidir. Bu işlem için literatürdeki 5 farklı süperpiksel algoritması kullanılmıştır. İkinci çalışma ise gürültülü imgelerin süperpiksel performanslarının incelenmesidir. Gürültülü imgeler, orijinal imgelere farklı gürültü tiplerinde farklı gürültü seviyelerinde gürültü ekleyerek yapay imgeler

oluřturularak elde edilmektedir. Bu alıřmada gauss gürültüsü, tuz&biber gürültüsü ve bulanıklařtırma olmak üzere 3 tip gürültü tipi üzerinde alıřılmıřtır. Ayrıca gauss gürültüsü 10 farklı seviyede, tuz&biber gürültüsü 4 farklı seviyede ve bulanıklařtırma 4 farklı seviyede uygulanmıřtır. alıřmanın son ařaması ise gürültülü imgelerin performansının iyileřtirilip iyileřtirilemeyeceğinin incelenmesi için gürültülü imgelere iki farklı filtre uygulanmıřtır. Filtreli imgelerde de süperpiksel performansları incelenmiřtir. Süperpiksel algoritmalarının orijinal imge, gürültülü imge ve filtreli imgeler üzerinde performansı başarı metrikleri ile karřılařtırılmıřtır.



2. KAYNAK ÖZETLERİ

Süperpikseller, özellikle imge ve video bölütleme uygulamalarında kullanılan; bir imgeye ait renk, yoğunluk ve doku özellikleri benzerlik gösteren piksellerden oluşan görsel yapılardır. Süperpiksel algoritmaları nesne tanıma [Mori vd. (2004), Li vd. (2012)], imge bölütleme [Ren ve Malik (2003), Peng ve Zhang (2011)], kenar tespiti [Levinshtein vd. (2010)], sınıflandırma [Liu vd. (2013)] gibi birçok görü sisteminde kullanılmaktadır. Süperpikseller oluşturulurken bazı özelliklere sahip olması istenmektedir. Bunlardan ilki imgedeki nesne sınırı ile oluşturulan süperpikselin sınırının uyumlu olması istenilmektedir. Bir diğer özellik ise süperpiksel algoritmaları bilgisayarlı görü sistemlerinde kullanılırken ön işlem olarak sıklıkla tercih edildiğinden işlem karmaşıklığının az olması ve işlemsel süre açısından hızlı olması beklenilmektedir. Bir başka özellik ise bir süperpiksel içerisinde bir nesnenin yer alması istenilmektedir ve böylece tespit, sınıflandırma gibi uygulamalarında kullanıldığında işlem kolaylığı sağlamaktadır. İnsanlar gördükleri imgelerdeki süperpikselleri kolayca ayırt edebilme yeteneğine sahipken, otomatik görü sistemleri ve bilgisayar yazılımları için bu işlemi yapmak oldukça zordur. Bu yüzden süperpiksel algoritmaları imge işleme sırasında farklı kullanım alanlarına sahip olduğundan tek bir başlık altında toplamak mümkün değildir. Bu yüzden kullanılan alana bağlı olarak çeşitli yöntemler geliştirilmiştir. Kullanılan metotlar imgedeki piksel gruplarını oluşturmak için algoritmalar su seti tabanlı, yoğunluk tabanlı, graf tabanlı, kontur tabanlı, yol tabanlı, kümeleme tabanlı ve enerji tabanlı algoritmalar olarak ana başlıklar altında toplanabilmektedir. Aşağıda bu algoritmalar detaylı bir şekilde anlatılacaktır.

2.1. Su Seti Tabanlı Algoritmalar

Su seti tabanlı algoritmalarının temel amacı set çizgilerini bulmaktır. Set kavramı, bir imgeyi iki uzamsal koordinat ve yoğunluk olmak üzere üç boyutlu olarak göstermeye dayalıdır. Bir deliğin her bir minimum bölgede yerleştirildiğini ve bütün topografyanın bir biçimli oranda aşağıdan itibaren delikler boyunca su yükselmesine izin verilerek taşırıldığı farz edilir. Aynı akaçlama havzalarındaki yükselen su üst tarafta birleşmeye başladığında, birleşmeyi engellemek için bir

su seti yapılır. Su bentlerinin tepeleri sadece su çizgisi yukarısında görünür olduğunda, taşma son konuma ulaşmış demektir. Su seti sınırları setlerin bölme çizgilerini verir. Bu yüzden setler, set bölütleme algoritması ile çıkarılacak olan sınırlardır. Set uygulamalarından birisi, nesnelere arka plandan çıkarmaktır. Yoğunlukta küçük değişiklikler ile karakterize edilmiş bölgeler küçük gradyan değerlerine sahiptir. Bu nedenle, imgenin doğrudan kendisine değil, imgenin gradyanına set bölütleme uygulaması tercih edilir (Gonzalez, 1992).

Vincent ve Soille (1991), sayısal gri seviyeli imgelerde su seti hesaplamak için hızlı bir algoritmadır. Mevcut algoritma suya daldırma işlemi benzetimine dayanmaktadır ki bu işlemde imge içerisinde suyun akışı bir piksel dizisi kullanılarak etkili bir şekilde simule edilir. Bu çalışmada, tüm pikseller üzerinde tarama işlemi gerçekleştirilmektedir ve iki adımdan oluşmaktadır. İlk adım imgedeki gri seviyeli değerlerin artan sıraya göre sıralanması işlemidir. Sıralama adımında piksellerin sıralanması için imgenin iki kez taranması gerekmektedir. İkincisi ise taşma adımındır ve her piksel ortalama üç kez taranır. Bu adımda bütün pikseller sıralandıktan sonra bir eşik değeri belirlenir ve etiketleme işlemi burada gerçekleşir. Öncelikle her bir piksele başlangıç etiketi atanır. Daha sonra eşik değerine göre yeni etiketler atanarak güncellenir ve üçüncü taramada yeni minimum değerleri ortaya çıkarsa etiket güncellemesi yapılarak imge bölütleme işlemi gerçekleştirilir. Bu algoritmanın imge bölütlemelerine ilişkin uygulamaları manyetik rezonans (MR) görüntüleri ve sayısal yükseklik modelleri için sunulmuştur. Ayrıca üç boyutlu bir havza örneği de sağlamaktadır. Bu çalışmada çalışma süresi bakımından oldukça hızlı olmasına rağmen oluşturulan bölütler farklı boyutlarda ve düzensiz yapıdadırlar.

Benesova ve Kottman (2014), bu çalışma bir imgeyi süperpiksellere morfolojik bir yaklaşımla bölmek için hızlı bir yöntem sunmaktadır. Bu çalışma nesne tespiti çalışmasının bir ön işleme adımı olarak kullanılması amaçlandığından hızlı çalışması, küme içindeki varyans değerinin düşük ve kümeler arası varyans değerinin yüksek olması, imge uzayında kümelerin uzamsal olarak tutarlı olması ve sınırların düzenli olması istenilmektedir. Morfolojik imge yeniden yapılandırma işlemini, imgedeki alakasız uzamsal yerel aşırı yoğunlukları ortadan

kaldırmak ve böylece alakasız kenarları kaldırmak için kullanılmaktadır. Morfolojik işlem sonunda gradyan kenar bilgisine sahip imge oluşmaktadır. Morfolojik su seti bölütlemesi için tohumlar üretilmektedir. Bu tohumların süperpiksellerin düzenli olarak dağılımına ve çıkıntı kenarlara uyması gerekmektedir. Su seti algoritması tohumların düzenli dağılımına dayanmaktadır. Her bir tohum ilk olarak düzenli ızgaralar tarafından verilen dikdörtgen alanın yerel minimumlarına yerleştirilir. Gradyan imgesinde tohumlar morfolojik taşma ile büyümekte fakat düzenli ızgaraların dikdörtgensel alanları ile sınırlandırılmıştır. Gradyan bilgisi ve düzenli oluşumu arasında denge oluşturmak için bir eşik değeri belirlenmektedir. Tohumların kontrol edilebilmesi için tohum ve tohumun komşuları arasında renk değerlerinin farkına bakılmaktadır. Bu fark işlemi de tohumlar arasındaki benzerliği ölçmektedir. Gerçek zamanlı veya neredeyse gerçek zamanlı uygulamalar için uygun bir algoritmadır.

Machairas vd. (2015), süperpiksel üretmek için su seti dönüşümü uygulanan bir strateji önerilmiştir. Süperpiksel düzenliliği ile nesne sınırlarına uyma arasında ayarlanabilir bir dengeye ulaşmak için uzamsal olarak düzenlenmiş bir gradyan kullanılmaktadır. Algoritmada ilk olarak imgenin gradyanı hesaplanmaktadır ve imge düzenli olacak şekilde hücrelere ayrılmaktadır. Her hücrenin merkez noktası minimum gradyan değeri hesaplanarak bulunmaktadır. Mesafe fonksiyonunun yardımı ile gradyanın uzamsal olarak güncellemesi gerçekleştirilmedi. Uzamsal olarak düzenlenmiş gradyan su seti dönüşümüne uygulanır, merkezlerden başlayarak taşma işlemi gerçekleştirilir ve su pikseli oluşturulur. Oluşturulan süperpikseller hem nesne sınırlarına uyma hem de düzenli süperpiksel yapıları oluşturulması bakımından doğruluğu gösterilmiştir. İşlemsel karmaşıklıkları doğrusal ve çalışma süresi bakımından hızlı bir çalışmadır.

2.2. Yoğunluk Tabanlı Algoritmalar

Yoğunluk tabanlı algoritmalar, yoğunluğu hesaplanmış imgede mod aramayı gerçekleştirir ve imgedeki her bir piksel ilgili moda atanır. Bu tip algoritmalar

genellikle süperpiksel sayısını veya kompaktlığını kontrol edemez. Bu nedenle yoğunluk tabanlı algoritmaları aşırı bölütleme algoritmaları olarak da sınıflandırılır.

Comaniciu ve Meer (2002), bu çalışmada yoğunluk kestirimi metotlarından en popüler olan çekirdek yoğunluk kestirimi kullanılmıştır. İmgeler özellik (renk, doku) uzayına dönüştürülerek işlem yapılmıştır. Dönüştürülen uzayda yerel maksimum bulmayı amaçlar. Algoritmada, her bir veri noktası için veri kümesinin olasılık yoğunluk fonksiyonunun yakındaki tepe noktasıyla ilgilenir. Algoritma kernel veya pencere genişliği (bant genişliği) belirlenmesiyle başlar. Verideki her bir nokta için o noktanda içinde olacağı şekilde pencere yerleştirilir. Pencere genişliğinin boyutuna karar verilmesi önemlidir. Çünkü pencere boyutu eğer küçük seçilirse veri sayısı az olacağından doğru bir bölüt oluşmayacaktır. Aynı şekilde pencere boyutu büyük seçilirse bir pencere yapısının içerisinde çok fazla veri olacağından yine doğru bir bölütleme olmayacaktır. Bu penceredeki veri noktalarının ortalama değeri alınır ve hesaplanan ortalama değere göre yeni merkezin konumu belirlenir. Bu işlemler yakınsama gerçekleşinceye kadar devam etmektedir. Burada yakınsamadan anlatılmak istenen merkez noktalarının bir sonraki adımda değişmemesidir. Çalışmada hem gri seviyeli imgeler hem de renkli imgeler üzerinde çalışılmıştır.

Vedaldi ve Soatto (2008), çalışmalarında ortalama kayma ve medoid kayma kümeleme algoritmalarının sınıflandırma problemlerinde başarısızlığını çözmek için basit, hızlı ve etkili yeni bir kümeleme algoritması sunmuşlardır. Algoritma genel olarak düşünüldüğünde iki adımdan oluşmaktadır. İlk adım, her bir pikselin yoğunluklarının Parzen pencere kestiriminin iteratif bir şekilde oluşturulmasıdır. Parzen pencereleri, yerel yoğunluğunu tahmin ederek yaklaşık olasılık yoğunluğunu bulur. Ölçek parametresi yardımıyla pencere boyutu ayarlanmaktadır. İkinci adım ise yoğunluk kestirimi fonksiyonunun değerini arttırabilmek için pikseller kendilerine en yakın pikseller ile bağlanarak bir ağaç yapısı oluşturmaktadırlar. Bu ağaç yapısı belirli bir eşik değerine kadar piksellerin birbirine bağlanması ile oluşturulmaktadır. Yani eşik değerinden daha büyük ağaçların dalları kırılarak yeni kümeler oluşturulmaktadır. Bu çalışma

eksik ve aşırı bölütleme arasında bir denge kurabilmektedir. Ortalama kayma algoritması ile karşılaştırıldığında daha hızlı bir performans sergilemektedir.

2.3. Graf Tabanlı Algoritmalar

Graf tabanlı algoritmalarda graf teorisinden yararlanılmaktadır. Bu teoriye göre imgedeki her bir piksel düğüm olarak adlandırılırken, düğümlerin birbiriyle bağlanmasını sağlayan bağlantılara da kenar adı verilmektedir. Oluşan her kenar bağladığı iki düğümün benzerliği ile ağırlıklandırılmaktadır. Düşük ağırlığa sahip olan kenarlar silinerek kümeleme işlemi gerçekleştirilmektedir. Böylece benzer pikseller aynı kümede yer almaktadır. Benzerlik ölçümü, iki öznitelik vektörü arasındaki uzaklığa bağlı olarak hesaplanmaktadır. Denklem 2.1' de σ parametresi uzaklığın benzerlik ölçüsüne etkisi ayarlamaktadır. x_i ve x_j ise farklı iki düğümü ifade etmektedir. Graf tabanlı algoritmaların dezavantajları kümeleri eşit boyutlu yapmaya yatkın olması, yüksek hesaplama maliyetine ve işlem yüküne sahip olması gibi özellikler sıralanabilir.

$$w(i, j) = \exp\left(-\frac{1}{2\sigma^2} \text{dist}(x_i, x_j)^2\right) \quad (2.1)$$

Shi ve Malik (2000), görü sistemlerinde algısal gruplama problemini çözmek için yeni bir yaklaşım önermişlerdir. Yerel öznitelik ve imge verisindeki yerel noktaların uyumuna odaklanmak yerine imgenin genel öznitelikleri kullanılmıştır. İmge bölütleme bir graf problemi olarak ele alınmıştır ve yeni bir ölçüt olarak normalleştirilmiş kesme önerilmektedir. Normalleştirilmiş kesme kriteri farklı gruplar arasında benzemezliği incelerken aynı grup içerisinde benzerliği ölçmektedir. Bu kriterin optimize edilmesi için genelleştirilmiş bir özdeğer problemine dayanan etkili bir hesaplama tekniği kullanılmıştır. Bu yaklaşımda gruplama graf teorisinin formülü ile ilişkilendirilmiştir. Ağırlıklı yönsüz graf $G = (V, E)$ düğümler ve bu düğümleri birbirine bağlayan kenarlardan oluşmaktadır. Her bir kenarın ağırlığı $w(i, j)$, i ve j düğümlerinin benzerliği ile ölçülmektedir. Graf iki ayrı kümeye ayrılmaktadır. Bu kümeler $A, B, A \cup B = V, A \cap B = \emptyset$ şeklindedir. İki parça arasındaki benzemezlik derecesi

kenarların toplam ağırlığı hesaplanarak bulunmaktadır. W aynı zamanda maliyet fonksiyonudur. Maliyet fonksiyonu bağlı olarak özvektör ile en küçük özdeğer $(D-W)x = \lambda Dx$ formülü kullanılarak çözülür. Özvektör ile ikinci en küçük özdeğer kullanılarak graf ikiye bölünür. Eğer devamında segmente etmek gerekirse parçalar yeniden belli bir eşik değerine kadar bölünmektedir.

Shen vd. (2014) çalışmalarında tembel rasgele yürüyüş (lazy random walk, LRW) kullanarak yeni bir imge süperpiksel bölütleme yaklaşımı önermektedir. Önerilen süperpiksel bölütleme algoritması rasgele yürüyüş (random walk, RW) tabanlı bir algoritmadır. Fakat orijinal RW pikseller arasında pikseller arasında yerel ilişkilerine ve ayrıca piksel ve ona ait tohum pikselleri arasındaki ilk olasılıklarına bağlıdır. Bu algoritma ile oluşturulan süperpikseller düzgün dağılımlı olmayan ve düzensiz süperpikseller oluşturulmaktadır. LRW ile tohumlar ve tüm pikseller arasında bölgesel ilişkiye dayanarak kompaktlık kısıtı ile yeni süperpiksel algoritması oluşturulmuştur. LRW ile öz döngü karmaşık doku bölgeleri ve güçsüz sınırları çözülmektedir. Bu algoritma başlangıç tohumlarının pozisyonlarının hassasiyetinden zarar görebilir. Bu kısıtlamanın üstesinden gelmek ve performansı arttırmak için, bir enerji optimizasyonu çerçevesi getirerek yeni bir süperpiksel optimizasyon yaklaşımı geliştirilmiştir. Bu enerji fonksiyonu tohumların konumunu optimize etmekte ve daha homojen süperpikseller oluşturulması için büyük süperpikselleri bölmektedir. Önerilen algoritma iki adımdan oluşmaktadır. İlk adım LRW algoritması ile başlangıç tohumlarının belirlenmesidir. Başlangıç tohumları belirlenirken graf teorisindeki düğümlere ve kenarlara dönüştürülerek bulunmaktadır. İkinci adım ise enerji fonksiyonudur. Denklem 2.2'de verilen enerji fonksiyonu da iki terimden oluşmaktadır. Bunlardan ilki veri terimidir ve bu terim süperpiksellerin homojenliğini ve tohum konumlarının güncellenirken düzenli bir şekilde sıralanmasını sağlamaktadır. İkinci terim ise düzlük terimidir. Bu terimde karmaşık dokulu büyük süperpikselleri daha homojen süperpikselleri oluşturmak için kullanılmaktadır.

$Area(S_l)$ süperpiksel alanını, $Area(\bar{S})$ süperpiksellerin ortalama alanını tanımlamaktadır. \tilde{W}_x süperpiksel etiketleri için penaltı fonksiyonudur. $\tilde{W}_x = e^{-CT(c_l^n, x)/\beta}$ şeklinde tanımlanmaktadır. β ise normalizasyon faktörüdür. $CT(c_l^n, x)$ ile işlem süresi ifade edilmektedir. c_l tohumu ve x pikselleri ifade etmektedir. Algoritmanın

doğruluğu test etmek için Berkeley veri kümesindeki imgeleri üzerinde EBH, EBD ve SÖ metrikleri incelenmiştir.

$$E = \sum_l (Area(S_l) - Area(\bar{S}))^2 + \sum_l \tilde{W}_x CT(c_l^n, x)^2 \quad (2.2)$$

Felzenszwalb ve Huttenlocher (2004), çalışmalarında bir imgenin bölgelere bölünmesi problemini ele almışlardır. İmgenin graf temelli bir yöntem ile iki bölge arasında kalan sınırı bulunması amaçlanmaktadır. İmge öncelikle imge teorisi yardımıyla düğüm ve kenarlara dönüşmektedir ($G = (V, E)$). İmgede düğümlerden oluşan bileşen adı verilen piksel grupları oluşturuluyor. Bu bileşenlerin 8 komşusu ile konum bilgisi (x, y) ve renk bilgisi (r, g, b) yardımıyla öklid mesafesi hesaplanmaktadır. Öklid mesafesinden yararlanılarak bileşenlerin belirlenmesi için iki ölçüm geliştirilmiştir. Bunlardan ilki graf teorisine dönüştürülen imgenin pikselleri yani düğümleri arasında ağırlık hesaplanmaktadır ve bileşenlerin en küçük yayılan ağaçtaki (minimum spanning tree, MST) en büyük ağırlığı ile tanımlanmaktadır (Denklem 2.3).

$$Int(C) = \max_{e \in MST(C, E)} w(e) \quad (2.3)$$

Bir diğer ölçüm ise bileşenler arasındaki farklılığı göstermek için oluşturulmuştur. Burada bileşenler arasındaki bağlantıların minimum değeri ile bulunmaktadır. Eğer Denklem 2.4'te verilen C_1 ve C_2 arasında bağlantı yoksa $Dif(C_1, C_2) = \infty$ şeklindedir.

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j) \quad (2.4)$$

Bu ölçümler ile bölge içerisindeki benzerliğin maksimum, bölgeler arasındaki benzemezliğin maksimum olması sağlanmaktadır. Bileşenler arasındaki farkı kontrol edebilmek için eşik fonksiyonu oluşturulmuştur. Denklem 2.5'teki formüller yardımıyla imge bölütleme işlemi gerçekleştirilmektedir. Bu çalışma gerçek ve sentetik imgeler üzerinde çalışılmıştır. Literatür taramasında bölütleme için öncü

çalışmalardan olmasına rağmen süperpiksel algoritmaları kategorisinde tam olarak yer almamaktadır.

$$D(C_1, C_2) = \begin{cases} true & \text{if } Dif(C_1, C_2) > M \text{ Int}(C_1, C_2) \\ false & \text{otherwise} \end{cases} \quad (2.5)$$

$$M \text{ Int}(C_1, C_2) = \min(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2))$$

Peng vd. (2015), çalışmalarında üst seviye enerji optimizasyonu yapısı kullanarak yeni bir süperpiksel çıkarımı algoritması önermişlerdir. Bu çalışmada başlangıç süperpiksel oluşumu için k-ortalamar algoritmasından yararlanılmıştır. Daha sonra üst seviye enerji fonksiyonu ile başlangıç süperpikselleri yeniden oluşturulmaktadır. Çalışmada kullanılan genel üst seviye enerji fonksiyonu üç bölümden oluşmaktadır (Denklem 2.6). Bunlardan ilki veri terimi, ikinci terim düzgünlük terimi ve üçüncü terim yüksek seviye terimidir. İlk iki terim kullanılarak enerji fonksiyonu oluşturulursa buna düşük seviyeli enerji fonksiyonu denilmektedir.

$$E(L) = \lambda_u E_{data}(L) + \lambda_p E_{smooth}(L) + \lambda_h E_{high}(L) \quad (2.6)$$

$E(L)$ enerji fonksiyonunu, L ise süperpiksellerin etiketlerini tanımlamaktadır. $\lambda_u, \lambda_p, \lambda_h$ sırasıyla tekli, çiftli ve yüksek seviyeli ağırlık parametreleridir. İlk terim veri terimi olup süperpiksellerin homojenliğini kontrol etmektedir. İkinci terim düzgünlük terimi olup süperpiksellerin şekilsel olarak kompaktlığını inceleyip pürüzsüz düzgün bir süperpiksel oluşumu için kullanılmaktadır. Üçüncü terim ise yüksek seviye terimidir ve oluşturulan süperpiksellerin imgenin sınırları ile uygunluğunu sağlayabilmesi için kullanılan metriktir. Çalışmanın diğer süperpiksel algoritmaları doğruluğunun karşılaştırılabilmesi için SÖ, EBD ve EBH metrikleri kullanılmıştır. Bu metrikleri test edebilmek için de Berkeley veri kümesi kullanılmıştır.

Veksler vd. (2010), çalışmalarında süperpiksel oluştururken graf tabanlı yöntemlerden graf kesme algoritmasını optimize ederek nesne bölütleme için kullanmışlardır. Bu çalışmada, oluşturulan süperpiksellerin düzenli yapılarda olması amaçlanmış ve bu

yapıları oluşturabilmek için enerji minimizasyonundan yararlanılmıştır. Süperpiksel algoritmalarının genel problemi olan etiketleme problemini çözebilmek için veri kısıtı ve düzlük kısıtı olmak üzere iki kısıttan yararlanılmıştır. Bu kısıtlardan veri kısıtı imgedeki piksellerin hangi etikete atanacağını belirlerken düzlük kısıtı ise komşu piksellerin farklı etiketlere atanmasını belirlemektedir. Bu çalışmada enerji fonksiyonu aşağıdaki gibi verilmiştir.

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{\{p,q\} \in N} \omega_{pq} \cdot V_{pq}(f_p, f_q) \quad (2.7)$$

Eşitliğin ilk toplam ifadesi veri kısıtını, ikinci toplam ifadesi düzlük kısıtını ifade etmektedir. λ , veri ve düzlük kısıtı arasındaki ağırlık parametresidir. Bu çalışmanın sonucunda kompakt süperpikseller ve sabit yoğunluklu süperpikseller oluşmaktadır. Farklı iki süperpiksel gruplarının oluşmasının nedeni veri kısıtının içeriğin değişmesinden kaynaklanmaktadır. Bu çalışmada Berkeley veri kümesi kullanılmıştır ve oluşturulan süperpikselleri test edebilmek için SÖ ve EBH metrikleri hesaplanmıştır.

Liu vd. (2014), çalışmalarında graf tabanlı kümeleme işlemi için yeni bir amaç fonksiyonu oluşturmuşlardır. Bu amaç fonksiyonu iki bileşenden oluşmaktadır. Bunlar graftaki rasgele yürüyüş işleminin entropi oranı $H(A)$, kompakt ve homojen kümelerin oluşumu ve dengeleme terimi $B(A)$ ise yaklaşık aynı boyutlarda süperpiksel oluşumunu sağlamaktadır. Başlangıç süperpiksel oluşumu için graf teorisinden yararlanılmaktadır. $G = (V, E)$, V değerleri her bir pikseli ifade ederken, E değerleri ise pikseller arasında bağlantı kuran kenar yapılarını ifade etmektedir. Algoritma, bahsi geçen iki fonksiyondan oluşan amaç fonksiyonunu maksimize etmek üzere geliştirilmiştir. Bilindiği üzere, bir veri kümesi için entropi değerinin düşük olması veri kümesinin homojen bir yapıya sahip olduğunun göstergesidir. Bu nedenle, entropi oranı kümeleme (Entropy rate clustering, ERC) algoritması entropi değerleri mümkün olan en düşük değerdeki, dolayısıyla en homojen süperpikselleri elde etmeyi amaçlamaktadır. Rasgele yürüyüş yöntemi yardımıyla piksel yoğunluklarına bağlı olarak başlangıç süperpiksel bölge sınırları oluşmaktadır. Oluşan başlangıç süperpikseller entropi değeri yardımıyla homojen ve kompakt yapı oluşumu sağlanmaktadır. Daha sonra homojen ve kompakt süperpikseller yaklaşık olarak aynı

boyutlarda olabilmesi için denge terimi ile süperpiksellerin son hali oluşmaktadır. Denklem 2.8’de verilen amaç fonksiyonundaki λ terimi kompakt ve homojen süperpiksel yapısı ile yaklaşık aynı boyutlarda süperpiksel oluşumu için arasındaki ağırlık katsayısıdır.

$$\max_A (H(A) + \lambda B(A)) \quad (2.8)$$

2.4. Kontur Tabanlı Algoritmalar

Kontur tabanlı yaklaşımlar genellikle kenar algılamak için oluşturulan yöntemlerden biridir. Kenarları bulabilmek için imgede önceden belirlenen sayı ile küme merkezleri oluşturulmaktadır. Belirlenen merkez noktalarından başlayarak bir maliyet fonksiyonu ile konturlar genişlemektedir. Gelişen konturlar süperpikselleri oluşturmaktadır fakat bu gelişme aşamasında etiketlenmemiş pikseller kaldığı için son işleme gerek duymaktadır.

Levinshtein vd. (2009) çalışmalarında, süperpiksel olarak adlandırılan bir imgenin yoğun bir yansımasını hesaplamak için geometrik akış temelli bir algoritma tanımlanmaktadır. Algoritma bir taraftan lokal imge sınırlarını korurken bir taraftan da kompaktlık kısıtı ile az bölütlemeyi sınırlandırmaktadır. Çalışmanın temel fikri; süperpiksel hesaplamasını etkili bir şekilde çözülebilir bir geometrik akış problemine indirgemektedir. Algoritma beş ana prensibe dayanmaktadır. Bunlardan ilki düzgün boyut ve kapsama alanıdır. Süperpiksel bölütleme bir imgeyi bölütlerken oluşan bölütlerin boyut ve şekil bakımından hemen hemen eşit alanlarda oluşumunun incelenmesi durumudur. Çalışmada her bir tohum süperpiksele karşılık gelmektedir ve düzgün dağıtılan tohumların başlangıç kümeleri genişleyerek bir geometrik akış tasarlanarak gerçekleşmektedir. İkincisi ise bağılıktır ve her bir süperpiksel piksel kümelerinin birlikteliği ile gösterilmektedir. Bu çalışmada genişleme tabanlı akış, seviye kümesi (level set) uygulaması ile birleştirilmiştir. Bir diğer prensip kompaktlıktır. Bölgesel kenar bilgisi bilinmediğinde süperpikseller kompakt yapısından çıkarım yapılmalıdır. Merkez noktasından dairesel olarak genişleme başlar ve süperpiksel sınırlarının yerleri hakkında öncü bilginin olmadığı

varsayılmaktadır. Kompaktlığı maksimum yapmak için merkez noktasından başlayarak dışa doğru genişleyen ve sabit yoğunlukta bir bölge oluşumuna kadar devam etmektedir. Dördüncü prensip ise pürüzsüzlüktür. Büyüme durduğunda süperpiksel sınırları imgenin kenarları ile kesişmesi gerekmektedir. Geometrik akış formülasyonu için üç özellik gerekmektedir. Kenarlara yaklaştıkça sınır büyümeleri yavaşlamalı, kenarlar belirgin olmalı ve sınırlar pürüzsüz olmalıdır. Sonuncu ise çakışma olmama durumudur. Her bir piksele atanan etiket tek olmalıdır. Bu özellikler dahilinde algoritma adımları; başlangıç merkez noktalarının yerlerinin belirlenmesi, T süre boyunca sınırların genişlemesi, atanmayan piksel bölgelerinin iskelet yapısının oluşturulması ve bu iskelet yapısının güncellenmesi, hız güncellenmesi ve süperpiksel çıkarımı gerçekleştirilmesi ile sağlanır.

Byssens vd. (2014), süperpiksel algoritması oluştururken yerel imge sınırlarının korunması gerektiği ve eksik bölütlemenin sınırlandırılması gerektiği vurgulamaktadır. Algoritmada kullanılan potansiyel haritanın sabit olmadığı ve akış sırasında geliştiği bir noktadan başlayarak genişlemesine bağlı olarak bölgelerin oluşturulması önerilmektedir. Algoritmanın başlangıcı K adet tohum sayısının belirlenmesiyle başlamaktadır. Bu tohumlar imge üzerinde düzenli olacak şekilde $(S = \sqrt{N/K})$ yerleştirilmektedir. Burada N imgedeki toplam piksel sayısını ve S oluşan bir tohumun bir kenarının uzunluğunu temsil etmektedir. Tohum pikseli ve onun 4 komşu pikselinin ortalaması ve varyans değeri hesaplanmaktadır. Aynı hesaplama 3×3 pencere boyutundaki pikseller için de yapılmaktadır ve tohum varyans değerini düşüren piksele taşınmaktadır. İlk tohumların bu şekilde dağılması piksellerin homojenlik açısından dağınık olmasını önlemektedir. Başlangıç adımında oluşan ilk süperpiksel grupları hızlı yürüyüş yöntemi kullanılarak güncellenmektedir. Kullanılan bu yöntem teorik olarak karmaşık olsa da pratikte oldukça hızlı ve lineer bir davranış sergilemektedir. İlk tohumlar düzenli şekilde yerleştirildiği için imgenin bazı nesnelere uygun tohumları içermeyeceğinden iyi bir bölütleme gerçekleşmemiştir. Bunu düzeltmek için yeni tohumlar eklenerek yeniden hızlı yürüyüş yöntemi ile güncelleme yapılmaktadır. Süperpiksellerin istenilen sayıda oluşabilmesi için zayıf süperpikseller kaldırılarak yerine daha kompakt

süperpikseller oluşturulmaktadır. Burada zayıf süperpikseli tespit ederken bölge uyum grafi hesaplanmaktadır, graftaki her düğümün normalize hacmi hesaplanmaktadır ve hacmi en küçük olan süperpiksel zayıf süperpiksel olarak adlandırılmaktadır. Bu işlem sonucunda istenilen sayıda ve kompakt süperpikseller oluşturulmaktadır. Önerilen bu algoritma diğer süperpiksel algoritmaları ile hız açısından karşılaştırılabilir ölçüdedir. Ayrıca Berkeley veri kümesinden yararlanılmış ve algoritmanın doğruluğunu ölçmek amacıyla EBD, EBH, SÖ ve DO parametreleri kullanılmıştır.

2.5. Yol Tabanlı Algoritmalar

Yol tabanlı algoritmalarda istenilen süperpiksel sayısı kadar merkez pikselleri belirlenmektedir ve bu merkez piksellerinden belirli bir kritere göre pikseller boyunca yol bağlantıları ile imgeyi bölerek süperpikseller oluşturulmaktadır. Burada pikseller arasında yol bağlantıları kullanılırken dinamik programlama, en kısa yol problemi gibi algoritmalarından yararlanılmaktadır. Yol tabanlı süperpiksel algoritmalarında süperpiksel sayısı kolaylıkla kontrol edilebilmektedir fakat kompaktlık genellikle kontrol edilemez.

Fu vd. (2014) çalışmalarında, süperpikselerin oluşumu sırasında süperpiksel yapılarının düzenli olmasını amaçlamaktadır. Düzenli süperpiksel kavramını değerlendirmek için iki kritere bakılmaktadır. Kriterlerden ilki, süperpikseller piksellerin bilgisini temsil etmektedirler ve piksellerden oluşan bölgelerin birbirleriyle uyumlu olması gerekmektedir. İkinci kriter ise piksellerin yapı, homojenlik, mekânsal topolojisi gibi kendine ait özellikleri koruması gerekmektedir. Algoritma genel anlamda üç aşamadan oluşmaktadır. İlk aşama bağlantı noktalarının bulunması işlemidir. Giriş imgesi için kenar genlik haritasına ihtiyaç duyulmaktadır. Çalışmada gPb haritası kullanılmaktadır, çünkü nesne sınırlarında arka plan bölgelerine göre daha yüksek değer almaktadır. Sınır haritasının değer aralığı $(0,1]$ 'dir. Bağlantı noktalarının yoğunluğu süperpiksel sayısına bağlıdır. Süperpiksel sayısı, algoritmanın giriş parametresidir. Algoritmanın ikinci aşaması, bağlantı noktalarının yer değiştirmesi işlemidir. Bağlantı noktaları sınır haritasının maksimal genliğine göre optimal pozisyona hareket ettirilir. Komşu bağlantı noktalarıyla çarpışmayı sınırlandırmak için

bölgesel arama uzayı belirlenir. Her bir bağlantı noktasının optimal pozisyonda yer değiştirmesi için iki kısıt vardır. Kısıtlardan biri süperpiksellerin bağlantı noktaları nesne sınırları üzerinde olmasıdır ve diğer kısıt ise süperpiksellerin konumsal yapılarını koruması gerekmektedir yani başlangıçtaki ızgara yapıya yakın olması gerekmektedir. Algoritmanın son aşaması ise bağlantı noktalarının birleştirilmesi işlemidir. Burada komşu bağlantı noktaları dikey ve yatay olarak bağlanmaktadır. Bağlama problemi en kısa yol probleminden yararlanılarak çözülmektedir. En kısa yol problemi kullanılırken yönsüz graf teorisinden yararlanılmaktadır. Graf teorisinde piksellere düğüm ve bu düğümleri birbirine bağlantı kuran hatlara kenar adı verilmektedir. Kenarların ağırlığı, komşu düğümlerin ilişkisine bağlı olarak değişmektedir. Sonuç olarak oluşan süperpikseller ızgara benzeri yapılarda oluşmaktadır ve sınır uyumu en kısa yol yöntemi ile korunmaktadır. Izgara yapılarının kısıtlı olması ve sınırlı arama alanı sayesinde homojen süperpikseller oluşturulmaktadır.

Moore vd. (2008), yol tabanlı bir algoritma önerilmektedir. İmgelerdeki piksellerin sıralanışı düzenli bir topolojiden oluşmaktadır fakat süperpiksel oluşurken düzenli topolojiler bozulmaktadır. Bunun için oluşturulan süperpiksellerin düzenli bir örgü topolojisinde olması amaçlanmaktadır. Süperpiksellerin düzenli bir örgü topolojisini koruması için açgözlü algoritmasından yararlanılmıştır. Bu süperpiksel algoritmasının girişi sınır haritasıdır ve sınır haritalama iki piksel arasında anlamlı sınır sunan olasılık tabanlı iki boyutlu diziden oluşmaktadır. Sınır haritalama işleminde bir değişiklik yapılıyorsa değerinde güncelleme meydana gelir eğer değişiklik (ters çevirmek, yeniden ölçeklendirmek) yapılmıyorsa aynen kalmaktadır. Böylece sınır haritası 1 ve 0'lardan oluşur. Sınır değer haritalaması en düşük yani yolların ağırlığının minimum olarak bulunması gerekmektedir. Algoritmanın her adımında tüm imgede optimal yol aranmaktadır. Bu yol arama problemi dinamik programlama ile çözülmektedir. Dinamik programlama düğüm ve kenarların değerlerinin toplamını minimum yapan bir boyutlu yol bulmaktadır. Dinamik programlama yardımıyla oluşturulan bu süperpiksellerin kafes yapılarını kontrol etmek için süperpiksel sayısı, imgedeki yol genişlikleri ve yolların kıvrımlılık derecesinden yararlanılmaktadır. İmgeler test edilirken Berkeley veri kümesi kullanılmıştır ve

diğer algoritmalarla karşılaştırılırken açıklık değişimi (AD) ve ortalama doğruluk parametreleri kullanılmaktadır.

2.6. Kümeleme Tabanlı Algoritmalar

Kümeleme tabanlı algoritmalarda derinlik, konum ve renk bilgileri kullanılmaktadır ve merkez noktalarından başlayarak k-ortalamalar algoritması gibi kümeleme algoritmasından esinlenilmiştir. Sezgisel olarak oluşturulan kümeleme tabanlı algoritmalarda süperpiksellerin sayısı ve kompaktlığı kolaylıkla kontrol edilebilmektedir. Ayrıca bu algoritmalar iteratif olarak çalışmasına rağmen bağlılığı arttırmak için son işleme ihtiyaç duymaktadır.

Chen vd. (2017) çalışmalarında, doğrusal spektral kümeleme olarak adlandırılan ve düşük hesaplama maliyetleri ile doğal imgeler için hem yüksek sınır bağlılığı hem de görsel olarak kompaktlığa sahip süperpikseller üretebilen bir süperpiksel bölütleme algoritmasıdır. İmge pikselleri arasında hem alan yakınlığı hem de renk benzerliğini ölçen mesafe metriği kullanılarak imge bölütlemenin normalize kesme tabanlı (normalized cut, NCUT) formülasyonuna uyarlanmıştır. Normalize kesme çalışmasında olduğu gibi direk olarak öz tabanlı algoritma kullanmak yerine piksel değerleri yüksek boyutlu öznitelik uzayına haritalandırılmıştır. Bu özellik uzayında her bir noktanın uygun bir şekilde ağırlıklandırılmasıyla, ağırlıklandırılmış k-ortalamalar (k-means) ile normalleştirilmiş kesimlerin amaç fonksiyonları aynı optimum noktaları temsil etmektedir. Böylece önerilen öznitelik uzayında k-ortalamalar kümeleme algoritması yinelemeli olarak uygulanarak normalize kesme algoritmasının maliyet fonksiyonu optimize edilmektedir. Bu algorithmada her piksel on boyutlu bir öznitelik uzayının denklem 2.9'da verildiği gibi içindeki bir noktaya eşlenir, buradaki bölütleme için ağırlıklandırılmış k-ortalamalar algoritması kullanılmaktadır. Orijinal piksel uzayındaki normalize kesme algoritması ile on boyutlu öznitelik uzayındaki ağırlıklandırılmış k-ortalamalar algoritması arasındaki teorik eşitliğe bağlı olarak yerel olmayan bilgiler korunmaktadır. İmge normalize kesme algoritması için graf teorisindeki düğüm ve kenarlara dönüşmektedir. C_c ve C_s sırasıyla renk benzerliğini ve konum benzerliğini ölçmektedir.

$$\begin{aligned}
\phi(p) = & \frac{1}{w(p)} (C_c \cos \frac{\pi}{2} l_p, C_c \sin \frac{\pi}{2} l_p, 2.55C_c \cos \frac{\pi}{2} \alpha_p, \\
& 2.55C_c \sin \frac{\pi}{2} \alpha_p, 2.55C_c \cos \frac{\pi}{2} \beta_p, 2.55C_c \sin \frac{\pi}{2} \beta_p, \\
& 2.55C_s \cos \frac{\pi}{2} x_p, 2.55C_s \sin \frac{\pi}{2} x_p, 2.55C_s \cos \frac{\pi}{2} y_p, 2.55C_s \sin \frac{\pi}{2} y_p)
\end{aligned} \tag{2.9}$$

$$w(p) = \sum_{q \in V} W(p, q) = w(p)\phi(p) \cdot \sum_{q \in V} w(p)\phi(p)$$

Acanta vd. (2012) çalışmalarında, gradyan tabanlı süperpiksel algoritmaları arasında literatürde en çok kullanılan algoritmaların başında gelen algoritmayı geliştirmişlerdir. Bu algoritma, gradyan tabanlı süperpiksel algoritmalarının ortak özelliği olan başlangıç kümelerinin oluşturulması ve gradyan bilgisine bağlı olarak bu kümelerin yinelemeli güncellenmesi mantığına dayanmaktadır. İlk olarak oluşturulan başlangıç kümeleri için merkez noktaları 3x3 pencere boyutundaki noktalar arasındaki en düşük gradyan bilgisine göre belirlenmektedir. Daha sonra imgeye ait tüm piksellerin hangi süperpiksele ait olduğu atanır. Bu atama işlemi temel olarak k-ortalama algoritmasının mantığına dayanmaktadır. Burada her bir süperpikselin merkez noktası ile imgeye ait piksellerin arasındaki mesafeye bakılarak o pikselin en yakın olduğu süperpiksel belirlenir. Burada mesafe değeri için konum değerlerinden ve bu algortmada kullanılan $\{L,a,b\}$ renk uzayındaki renk bilgisinden yararlanılmaktadır ve (2.10) no'lu denklem kullanılarak hesaplanmaktadır. Bir pikselin etiketine mesafe değeri en küçük olan süperpikselin etiket değeri atanır. Atama gerçekleşikten sonra her bir süperpiksel merkezi, süperpiksel içerisindeki piksellerin ortalaması alınarak güncellenir ve bu güncelleme işlemi artık hata değerinin belirlenen eşik değerinden küçük olma şartını sağlayana kadar yinelenir. Artık hata, yeni merkezlerle eski merkezler arasında öklid uzaklığının hesaplanması ile elde edilir. Ancak tüm piksellerin tek tek kontrol edilmesi işlem yükünü arttırdığından SLIC algoritmasında işlem yükünün azaltılması için k-ortalama algoritmasında yeniden düzenlenmiştir ve imgenin tüm uzayının kontrol edilmesi yerine $2S \times 2S$ arama uzayı kullanılmaktadır. Burada S başlangıç adımında belirlenen süperpikselin kenar uzunluğudur ve $S = \sqrt{N/k}$ formülü ile hesaplanmaktadır.

N imgenin tüm piksel sayısı, k ise belirlenen süperpiksel sayısına karşılık gelmektedir.

$$\begin{aligned}d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\D &= \sqrt{d_c^2 + m^2(d_s / S)^2}\end{aligned}\tag{2.10}$$

Denklem (2.10)' da d_c ve d_s oluşturulan süperpikseller ile imgeye ait piksellerin sırasıyla renk ve konum değerleri arasındaki mesafeyi göstermektedir. D ise etiket atama işleminde kullanılan mesafe ölçüsünü göstermektedir. Burada m , renk ve konum değerleri arasındaki ağırlık ölçüsüdür ve [1,40] aralığındadır.

Wang ve Wang (2012), imgedeki bölge sınırlarını koruyan ve bölgelerin tek biçimli olmasını sağlayan algoritma önerilmektedir. K-ortalama algoritması gibi uygulaması oldukça kolay bir kümeleme algoritmasıdır. Kümelerin sınır uzunluğu ile enerji fonksiyonu oluşturulmuştur. Sınır uzunluğunun ölçütü süperpiksel algoritmasının kompaktlığını göstermektedir. Algoritma imgeyi küçük parçalara ayırarak başlamaktadır. Daha sonra oluşturulan bu parçaların renk tabanlı merkezleri belirlenmektedir. Her piksel için piksel ve merkezler arasındaki kenar ağırlıklı mesafesi hesaplanarak en yakın merkeze atanmaktadır. Algoritma bölütleme performansı açısından başarılıdır fakat piksellerin konum bağlılığı düşüktür. Bu problemi çözmek için en yakın komşu arama algoritması (Looking nearest neighbors, LNN) ve bölütlere ayrılma tespiti (Detecting segment breaking, DSB) algoritmaları önerilmiştir. LNN, piksel ve merkezler arasındaki kenar ağırlıklı mesafeyi değerlendirmeye gerek olmadan hesaplama maliyetini azaltmaktadır. DSB ise pikseller arasındaki bağlılığı sağlamaktadır. Algoritmanın doğruluğunu test edebilmek için Berkeley veri kümesi kullanılmıştır. Ayrıca 0 ortalama ve 0.1 varyans değeri ile gauss gürültüsü ekleyerek süperpiksel performansı incelenmiştir. Algoritma diğer süperpiksel algoritmaları ile karşılaştırılırken SÖ ve EBH parametresi kullanılmıştır.

Zhou vd. (2015), geleneksel jeodezik uzaklık yerine yeni ikili jeodezik uzaklık önerilmiştir. Bu yeni uzaklık ile iki pikselin konumlar arasındaki uzaklığı hesaplandığında renkler arasındaki farklılık da hesaplanmaktadır. Jeodezik uzaklığa bağlı oluşturulan süperpiksellerin tohumları yüksek seviyeli süperpiksel oluşturabilmek ve enerji fonksiyonunun optimize edilmesi için oldukça önemlidir. İlk kümeler altıgen olarak oluşturulmaktadır ve tohumlar gradyan bilgisi kullanılarak belirlenmektedir. Tohum ve pikseller arasında ikili jeodezik uzaklık metriği hesaplanmaktadır. Bu ikili jeodezik uzaklık, bir tohum pikselden hedef piksele doğru birleşme yolundaki tüm piksellerin gradyanlarının toplamıdır. Gradyan tabanlı yöntemlerde tohumdan başlayarak süperpiksel metriklerini sağlayıncaya kadar devam etmektedir ve optimizasyon yardımıyla oluşturulmaktadır. Her pikselin 3×3 pencere boyutunda fast marching yöntemi ile hız fonksiyonu oluşturulmaktadır ve güncelleme enerji fonksiyonu ile gerçekleştirilmektedir. Algoritmanın durabilmesi için iki ardışık iterasyon arasındaki enerji değişimi 2 olarak belirlenmiştir ya da iterasyon sayısı 20'ye ulaşması gerekmektedir. Bu algorithmada Berkeley veri kümesi kullanılmıştır. Ayrıca algoritma doğruluğunun test edilmesi için EBH, SÖ ve EBD kullanılmıştır.

Zhang vd. (2017), oluşturulan süperpiksellerin sınır uyumluluğu, yoğunluk bakımından homojenliği ve kompaktlık özelliklerini dengelemek için bir mesafe fonksiyonuna dayanan yeni bir süperpiksel bölütleme algoritması önermektedir. Bu çalışmada ideal bir süperpiksel için en önemli özelliklerden biri imgedeki nesne sınırı ile süperpiksel kenarının hizalanması savunulduğundan oluşturulan yeni mesafe ölçütü piksel ile süperpiksel tohumu arasındaki mesafeyi bulmak için tasarlanmıştır. Mesafe fonksiyonu sınır, yoğunluk ve kompaktlık olmak üzere üç terimden oluşmaktadır.

$$D(x,l) = \omega_b \times B(x,l) + \omega_i \times I(x,l) + \omega_c \times C(x,l) \quad (2.11)$$

Burada ilk terim sınır terimi, ikinci terim yoğunluk terimi ve üçüncü terim ise kompaktlık terimidir. $\omega_b, \omega_i, \omega_c$ farklı özelliklere göre oluşturulan fonksiyonun ağırlık katsayılarını göstermektedir. Sınır terimi; piksellerin gradyan değerlerinin hesaplanmasıyla imge sınırlarını tespit etmektedir. Yoğunluk terimi;

oluşturulan süperpiksellerin içerisinde piksellerin benzer renk yoğunluğa sahip olmasını sağlamaktadır ve Lab renk uzayını kullanmaktadır. Burada süperpiksellerin varyans değerleri de hesaplanmaktadır. Üçüncü terim kompaktlık terimi ise Öklid mesafesini kontrol etmektedir ve kompakt bir yapının düzgün altıgenden oluşacağı savunulmaktadır. Oluşturulan süperpiksellerin tohumlarının güncellenmesi için 3 sigma kuralından yararlanılmaktadır. Bu kural ile her süperpikselin ortalama ve standart sapma değeri hesaplanarak yeni bir güncelleme fonksiyonu oluşturulmaktadır. Çalışmada Berkeley veri kümesi ve Microsoft MSRC GRABCUT veri kümesi kullanılmıştır. Oluşturulan algoritmanın doğruluğunu test etmek için SÖ, EBH, DO ve EBH metrikleri hesaplanmıştır. Sonuç olarak oluşturulan süperpikseller ile nesnenin sınırlarına uyduğu sonucuna varılmaktadır.

2.7. Enerji Tabanlı Algoritmalar

Süperpiksel algoritmalarından bir diğeri olan enerji tabanlı süperpiksel algoritmaları iteratif olarak enerji formülasyonunun optimize edilmesiyle oluşmaktadır. Başlangıç süperpikselleri oluşturulurken imge düzenli kafeslere bölünerek meydana gelmektedir. Buna ek olarak pikseller enerji yardımıyla komşu süperpikselerde yer değiştirebilir. Enerji tabanlı süperpiksel algoritmalarında süperpiksel sayısı ve kompaktlığı kontrol edilebilmektedir.

Taşlı vd. (2015), çalışmalarında hem doğruluğu bakımından hem de hesaplama karmaşıklığı bakımından etkin bir süperpiksel ve süpervoksel çıkarma yöntemi sunmaktadır. Bu çalışma dört ana adımdan oluşmaktadır. Bunlardan ilki başlangıç süperpiksellerinin oluşturulmasıdır. Burada istenen süperpiksel sayısına göre imgeyi eşit boyutlarda bölme işlemi ile başlamaktadır. Her bir bölge başlangıçta dikdörtgen veya altıgen yapıda ve merkezleri eşit uzaklık olacak şekilde dağıtılmaktadır. Merkez noktalarından başlayarak özel bir enerji fonksiyonu yardımıyla süperpiksel sınırları düzenlenmektedir. Bu düzenleme iteratif olarak sınır ve yapı güncellemeleriyle başarılı sonuçlar elde edilmektedir. İkinci adım ise sınır güncellemesidir. Burada sınır pikselleri üzerinde açgözlü arama algoritması kullanılmaktadır. Sınır uyumu boyunca aday süperpiksellerdeki piksellerin benzer yapıda olması için maliyet fonksiyonu

minimize edilmektedir. Sınır pikselleri ile komşu süperpiksel adayları arasında arama yaptığı için hesaplama verimliliği başarılıdır. Sınır piksellerinin etiket ataması 8 komşusuna bakılarak yapılmaktadır. Piksel (p) ve süperpikseller (Q_i) ile gösterilmektedir ve aralarında bir fonksiyon oluşturulmuştur (denklem 2.12).

$$L(p) = \arg \min_{Q_i} (E(p, Q_i)), \quad (2.12)$$

$$Q_i \in N_p, i = 1: N$$

$L(p)$, p piksellerinin etiketleri göstermektedir. $E(p, Q_i)$ piksel ve süperpikseller arasında benzemezlik enerji fonksiyonunu, N komşu süperpiksellerin sayısını göstermektedir. $E(p, Q_i)$ renk bilgisi ve jeodezik mesafe bilgisi içeren bir enerji fonksiyonudur. Burada süperpiksellerin başlangıç dağılımı ile başlayıp sınır pikselleri en benzer komşu süperpiksele yeniden atanmaktadır. Tüm sınır piksellere bakıldığında, süperpiksel merkezleri ve ortalama renk değerleri o anki etiket bölgesine göre güncellenmektedir. Üçüncü adım ise yapı güncelleme adımudur. Süperpikseller sınır piksellerinin birleştirilmesi veya kaldırılması için yeniden hesaplanır. Değişiklikler ile sınırlara uygunluk ve kompakt süperpiksel oluşumu sağlanmaktadır. Sınır ve yapı güncellemeleri iteratif olarak durdurma kriterine kadar devam etmektedir. Son adım ise sonlandırma adımudur. Çalışmada farklı iterasyon seviyeleri için algoritma denenmiştir ve bu çalışma için en uygun iterasyon sayısı 10 olarak belirlenmiştir. İşlem sonrası adımında ise medyan filtre uygulanarak süperpiksel içindeki olası bağlantı noktalarını sınır piksellerinde uygulamaktadır. Sınır pikselleri çevresindeki farklı süperpiksel etiketlerine ait küçük parçaların oluşmasını engellemektedir. Çalışmalarında Berkeley veri kümesini kullanmışlardır. Süperpiksel algoritmalarının başarısını ölçmek için SÖ ve EBH metriklerini ve ayrıca bu iki metrik yardımıyla yeni ortalama doğruluk performansı metriği geliştirip kullanmışlardır.

Bergh vd. (2013), çalışmalarında imgedeki nesne piksellerinin aynı süperpiksele ait olarak gruplama işlemini amaçlamışlardır. Birçok süperpiksel algoritması renk homojenliği kısıtı kullanılarak hedef fonksiyonunun minimize edilmesiyle

oluşmaktadır. Bu optimizasyonları gerçekleştirmek için kesme veya genişleme algoritmaları uygulanmaktadır ve bu işlemler gerçek zamanlı uygulamalarda oldukça hesaplama maliyetini arttırmaktadır. Maliyeti düşük olan bir algoritma için tepe tırmanma algoritmasından yararlanılmıştır (hill-climbing algorithm). Bu çalışma başlangıç süperpiksellerinin oluşmasıyla başlamaktadır. Farklı boyutlarda piksel blokları ile başlanarak her bir bloğun histogramı hesaplanmaktadır. Blokların boyutları 2×2 veya 3×3 olarak alınmıştır. Algoritmada en büyük blok boyutu hedef süperpiksel boyutunun dörtte biridir. Böylece, süperpikseller en büyük blok boyutunun 2×2 bloklarının birleşimi olarak başlatılır. Güncelleme işlemleri yapılırken önce büyük bloklarla işlem yapılırken iterasyon sayısı arttıkça daha küçük bloklarla veya piksellerle güncelleme işlemi yapılmasının nedeni sınırları oluştururken detaylı olarak pikseller üzerinden işlem yapmaktır. Renk dağılım terimi ve sınır terimi hesaplanarak güncelleme için enerji fonksiyonu oluşturulmaktadır.

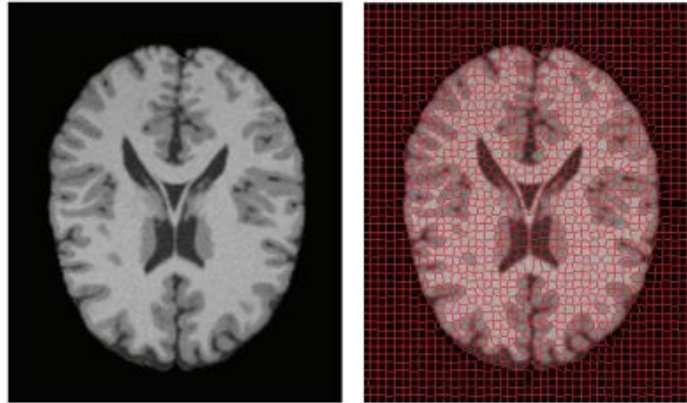
$$E(s) = H(s) + \gamma G(s) \quad (2.13)$$

$H(s)$ renk dağılımlarını inceleyen parametredir. Bu parametre ile süperpiksellerin olabildiğince homojen olması sağlanmaktadır. $G(s)$ sınır parametresidir. Süperpiksel sınırlarının yerel düzensizliğini dikkate alınarak hesaplanmaktadır. Bu parametre opsiyonel olarak kullanılmaktadır. Eğer sınır düzenliliği önemsenmiyorsa bu terimin kullanılması zorunlu değildir. γ ise sınır ve renk terimi arasındaki ağırlık parametresidir. Algoritma bu enerji fonksiyonu ile belirlenen iterasyon boyunca güncelleme yaparak süperpikselleri oluşturmaktadır. Çalışmada Berkeley veri kümesi kullanılmış olup SÖ, EBD, EBH ve CUE metrikleri ölçüm yapılmıştır.

2.8. Diğer Çalışmalar

Ji vd. (2014), çalışmalarında manyetik rezonans görüntüleme (magnetic resonance imaging, MRI) iyi bir beyin bölütleme sonucunu elde etmek için süperpiksel ve bulanık kümeleme(Fuzzy clustering, FCM) temelli yeni çok aşamalı bir bölütleme yöntemi önermektedirler. Bu çalışmada süperpiksel kullanılarak imge bölütlere ayrılmaktadır, FCM yöntemiyle süperpiksel

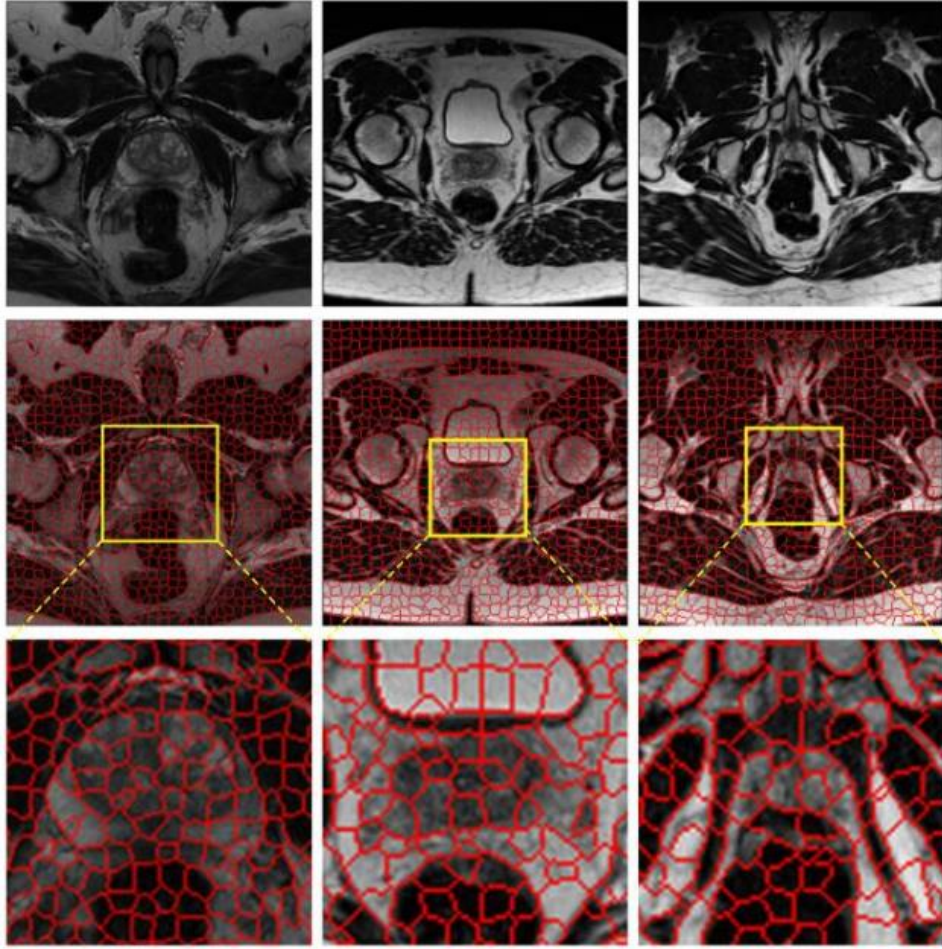
algoritması sonucunda oluşan süperpiksellere etiketleme işlemi gerçekleştirilmektedir. Algoritma genel olarak üç aşamadan oluşmaktadır. Bunlardan ilki süperpiksellere ayrılma işlemidir. Bu işlemi yaparken TP süperpiksel algoritmasından yararlanılmaktadır. TP algoritması başlangıç merkez noktalarının konumlarının belirlenmesiyle başlamaktadır. Daha sonra belirli bir T süre boyunca sınırlar genişlemektedir ve atanmayan piksel grupları ile iskelet yapısı oluşturulmaktadır. Son olarak hız güncellenmesi ile süperpikseller oluşturulmaktadır. İkinci adım ise derin bölütleme adıdır. TP süperpiksel algoritması ile elde edilen süperpiksellerin varyans değerleri hesaplanmaktadır. Böylece büyük varyans değerine sahip süperpikseller daha küçük atomik bölgelere dönüştürülmektedir. Son adım ise küme ve etiket süperpiksellerinin oluşması adıdır. Bu adımda, ilk iki adımda oluşan süperpiksellerin gri seviye ve konum bilgisine bağlı olarak FCM algoritması ile süperpiksellere etiketleme işlemi uygulanmaktadır ve aynı etiket bilgisine sahip süperpikseller birleştirilerek bölütleme sonucu elde edilmektedir. Çalışmalarında sentetik MRI imgeleri ve kesin referanslarının bulunduğu Brain Web veri tabanını kullanmışlardır.



Şekil 2.1. MR imgesi ve bölütlenmiş MR imgesi

Tian vd. (2016), çalışmalarında prostat yüzeyi elde etmek amacıyla süperpiksel tabanlı 3 Boyutlu (3B) graf kesme algoritması önerilmektedir. Prostat ve arka plan etiketlemesi yapılabilmesi için 3B süperpiksel ile graf kesme algoritması kullanılarak bir enerji fonksiyonu oluşturulmaktadır ve bu enerji fonksiyonunun minimizasyonu ile prostat yüzeyinin çıkarımı gerçekleştirilmektedir. Enerji fonksiyonu süperpiksel tabanlı şekil veri terimi, görünüm ve terimi ve

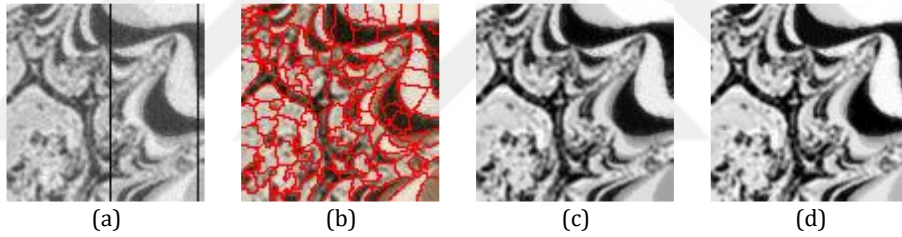
pürüzsüzlük terimlerinden meydana gelmektedir. Çalışma genel olarak graf kesme ve aktif kontur model içeren iki bölümden oluşmaktadır. Klasik aktif kontur modeli tıbbi imge bölütlemesi için kullanılmış olmasına rağmen, başlangıç yeterince iyi olmadığında yerel minima üzerine sıkışma problemi yaşanmaktadır. Bu problemin üstesinden gelebilmek için graf kesme algoritmasının çıktılarını aktif kontur modelinin girişi olarak kullanılmaktadır. Yani burada aktif kontur modeli graf kesme algoritması sonucunu güncellemek için kullanılmaktadır. Süperpikseller oluşturulurken Makni ve arkadaşlarının önerdiği gauss karma modeli kullanılmaktadır. Önerilen süperpiksel algoritmasında veri terimi ve pürüzsüzlük terimleri dikkate alınmaktadır. Burada veri terimi şekil ve gri seviye terimlerinden oluşmaktadır. Pürüzsüzlük terimi ise süperpiksel içi pürüzsüzlüğü ve süperpikseller arası pürüzsüzlük terimi olarak ikiye ayrılmaktadır. Buradan veri terimi ve pürüzsüzlük terimi hesaplamaları ile ilk süperpikseller oluşmaktadır. Daha sonra bu süperpikseller Chan ve Vese' in aktif kontur modeli kullanılarak süperpikseller güncellenmektedir. Çalışmalarında Emory Üniversite Hastanesi'nden T2 ağırlıklı 43 MR imgesi kullanılmıştır.



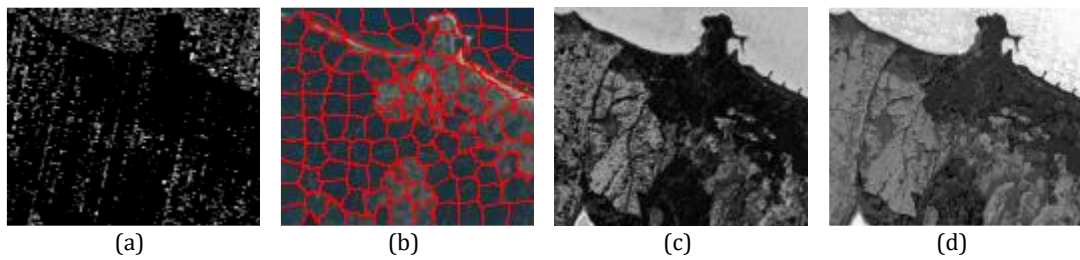
Şekil 2.2. Farklı prostat imgeleri ve bölütlenmiş prostat imgeleri

Ertürk (2016), çalışmasında hiperspektral imgelerdeki gürültü giderimi için süperpiksel tabanlı karışım gideri yaklaşımı önermektedir. Hiperspektral imgeler barındırdıkları yüksek miktardaki spektral bilgi sayesinde, sınıflandırma ve tespit gibi uygulamalarda sıklıkla kullanılmaktadır. Hiperspektral imgelerin elde ediniminden itibaren aktarım işlemleri boyunca çeşitli gürültülere maruz kalmaktadır. Karışım giderimi, hiperspektral görüntülerdeki piksellerin saf spektral imzalar ve bu imzaların bolluk oranları cinsinden ifade edilmesidir. Gürültü giderimi işlemi yapılırken hem uzamsal hem de spektral bilgi kullanılmaktadır. Karışım giderimi ile gürültü giderimi işlemi sayesinde gürültüsüz spektral imgelerin oluşturulması sağlanmaktadır. Karışım giderimi süperpiksellerden yararlanılarak gerçekleştirilmiştir. Süperpiksel kümeleme işlem karmaşıklığını azalttığı ve aşağı yukarı aynı boyutlarda olduğu için tercih edilmektedir. Eksik bölütleme hatası, sınır örtüşmesi diğer algoritmalarla karşılaştırıldığında daha iyi sonuç verdiği için SLIC algoritması kullanılmıştır.

Çalışma genel veri boyutu tespiti, süperpiksel kümeleme, karışım giderimi ve veriyi geri oluşturma olmak üzere dört adımdan oluşmaktadır. Veri boyutu tespiti Hysime ile gerçekleştirilmiştir. Süperpikseller elde edildikten sonra her süperpiksel için ortalama spektral vektörleri elde edilmektedir. Süperpikseller aracılığıyla elde edilen ortalama spektral vektörler N-FINDR veya zirve bileşen analizi (verteks component analysis – VCA) gibi son eleman çıkarımı yöntemlerine, HySime veri boyutluluğu çıktısı ile birlikte, girdi olarak verilmektedir. Süperpikseller ile elde edilen ortalama spektral vektörler orijinal piksel vektörlerine göre daha az gürültü olacağından dolayı bu aşamada son elemanlar daha başarılı şekilde çıkarılabilmektedir. Çalışmada hem sentetik veriler hem de gerçek veriler kullanılmıştır. Sentetik veriler için Fractal veri kümesi kullanılmıştır. Gerçek veriler için de EO-1 Hyperion verisi kullanılmıştır. Şekil 2.3 ve Şekil 2.4'te sırasıyla sentetik ve gerçek hiperspektral imgeleri için gürültü çıkarımı SLIC yöntemiyle gerçekleştirilmiştir.



Şekil 2.3. (a) Sentetik veri, (b) Süperpiksellere ayrılmış sentetik veri, (c) SLIC ve NFINDR yöntemi ile gürültü giderimi, (d) SLIC ve VCA yöntemi ile gürültü giderimi



Şekil 2.4. (a) Hyperion veri kümesi, (b) Süperpiksellere ayrılmış gerçek veri, (c) SLIC ve NFINDR yöntemi ile gürültü giderimi, (d) SLIC ve VCA yöntemi ile gürültü giderimi

Bu tez kapsamında kaynak özetlerinde yer alan süperpiksel algoritmaları hakkında genel bir özet oluşturmak için Çizelge 2.1 çizelgesi oluşturulmuştur.

Burada algoritmaların türü, kullanılan renk uzayları ve doğruluğunu test etmek için kullanılan başarı metrikleri verilmiştir.

Çizelge 2.1. İncelenen süperpiksel algoritmalarının listesi

Türü	Süperpiksel Algoritması	Yılı	Renk uzayı	Kullanılan Metrikler				
				EBD	SÖ	AD	EBH	DD
Su Seti Tabanlı	W	1991	Gri Seviyeli	-	-	-	-	-
Süperpiksel Algoritmaları	MSS	2014	RGB	-	✓	-	✓	-
	WP	2015	RGB	-	✓	-	-	✓
Yoğunluk Tabanlı	MS	2002	RGB	-	-	-	-	-
Süperpiksel Algoritmaları	QS	2008	RGB	-	-	-	-	-
	NC	2000	RGB	-	-	-	-	-
	GC	2004	RGB	-	-	-	-	-
Graf Tabanlı Süperpiksel Algoritmaları	SPSV	2010	Gri Seviyeli	-	✓	-	✓	-
	LRW	2014	RGB	✓	✓	-	✓	-
	ERC	2014	RGB	✓	✓	-	✓	-
	SHOE	2015	RGB	✓	✓	-	✓	-
Kontur Tabanlı Süperpiksel Algoritmaları	TP	2009	RGB	-	✓	-	✓	-
	ERGC	2014	LAB	✓	✓	-	✓	✓
Yol Tabanlı Süperpiksel Algoritmaları	SL	2008	RGB	-	-	✓	-	-
	RPS	2014	RGB	-	-	✓	-	✓
Kümeleme Tabanlı Süperpiksel Algoritmaları	SLIC	2012	LAB	-	✓	-	✓	-
	VCELLS	2012	LAB	-	✓	-	✓	-
Enerji Tabanlı Süperpiksel Algoritmaları	LSC	2017	LAB	✓	✓	-	✓	-
	SSBC	2017	LAB	✓	✓	-	✓	✓
Enerji Tabanlı Süperpiksel Algoritmaları	SEEDS	2013	LAB	✓	✓	-	✓	-
	CCS	2015	LAB	-	✓	-	✓	-

3. YAPILAN ÇALIŞMALAR

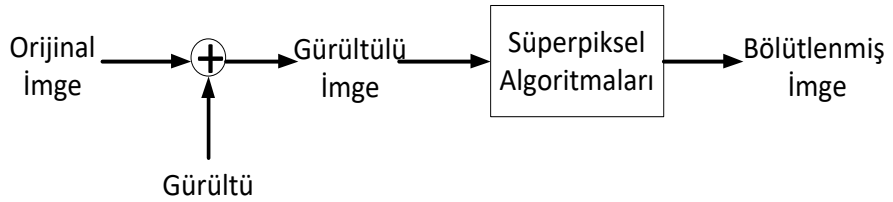
Süperpiksel algoritmaları günümüzde bilgisayarlı görü sistemlerinde, nesne tanıma ve tespiti sistemlerinde, görüntü işleme yöntemlerinde sıklıkla kullanılmaktadır. Bu sistemlerde kullanılan süperpikseller hem işlemsel açıdan hem de işlemsel süre bakımından düşük maliyetli olup görüntü işleme alanında önemli bir yere sahiptir. Ayrıca günümüzde veri transferi açısından değerlendirdiğimizde en çok imge gönderimi ve alımı yapılmaktadır. Haberleşme sistemlerinde veri gönderimi ve alımı sırasında istenmeyen etkilere yani gürültüye maruz kalabilir. Görüntü işleme işlemi hem alıcı hem de gönderici tarafında gerçekleşeceğinden uygulanan işlem sonucunda aynı olması istenmektedir. Fakat iletim sırasında maruz kaldığı gürültüden dolayı imgenin içeriği değişmektedir. Bunun yanı sıra tıpta kullanılan manyetik rezonans (Magnetic Resonance, MR), bilgisayarlı tomografi (Computed Tomography, CT), ultrason cihazlarından alınan sonuçların belirli bir bölgenin tespitinin yapılması yani o bölgenin tanımlanması gerekmektedir. Bu işlem yapılırken nesne tespitinde olduğu gibi biyomedikal görüntülerinin de tespiti için süperpiksel algoritmaları kullanılabilir. Fakat bu görüntüler kendi cihazlarına bağlı olarak gürültülü olarak meydana gelebilmektedir. Biyomedikal imgelerde belirli bir bölgenin doğru çıkartılması oldukça önemlidir. Yukarıda bahsedilen bu iki önemli durum için süperpiksel algoritmalarının gürültüye karşı bağışıklığı incelenmiştir. Süperpiksel algoritmalarının gürültü bağışıklığını incelemek için literatürde en çok kullanılan 5 farklı algoritma seçilmiş ve uygulanmıştır. Bu süperpiksel algoritmalarının performanslarını incelemek için seçilen veri kümesindeki 15 imgeye farklı gürültü tipleri ve farklı gürültü seviyeleri uygulanmıştır.

Bu tez çalışması 3 ana bölümden oluşmaktadır. Bunlardan ilki orijinal test imgelerin süperpikselle ayrılma işlemidir. Burada orijinal imgelere uygulandığında nesne sınırının bulunması, oluşan süperpiksellerin homojenliği değerlendirilmiştir. Şekil 3.1'de test imgesine 5 farklı süperpiksel algoritması uygulanarak orijinal imge üzerinde süperpiksel performansları incelenmiştir.



Şekil 3.1. Orijinal imgeler için bölütleme blok diyagramı

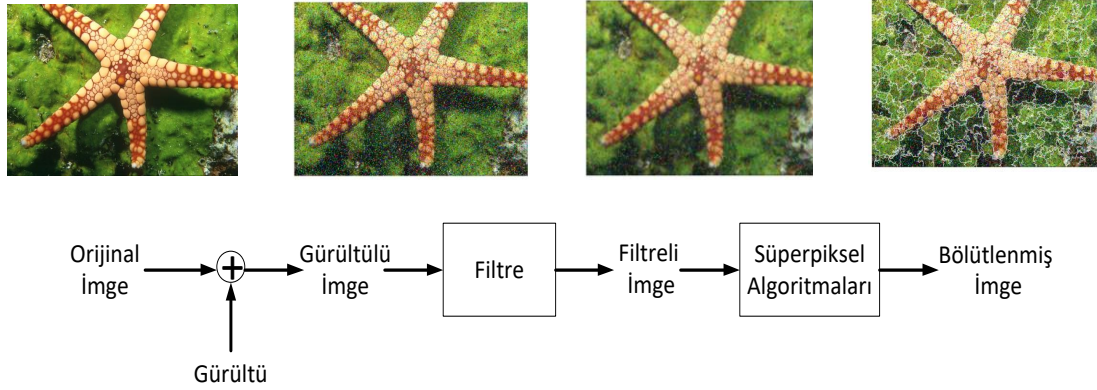
İkinci bölüm ise haberleşme kanallarının etkisiyle gürültülü imgeler oluşabilmektedir. Orijinal imgelere gauss, tuz&biber gürültüsü ve bulanıklaştırma etkisi eklenerek gürültülü imgeler oluşturulmaktadır. Orijinal imge ile gürültülü imge arasında yoğunluk değerlerinin farklılaşmasından dolayı bu gürültülü imgeler üzerindeki süperpiksel algoritmalarının performansının nasıl etkilendiği incelenmektedir ve Şekil 3.2'deki blok diyagramda gösterilmektedir.



Şekil 3.2. Gürültülü imgeler için bölütleme blok diyagramı

Orijinal imgelere eklenen gürültü ile imgenin yapısında değişiklik meydana gelmektedir. Gürültülü imgenin orijinal imgeye yakınsaması için görüntü işleme tekniklerinde sıklıkla kullanılan medyan ve wiener filtrelerden yararlanılmıştır. Bu filtreler yardımıyla yeni oluşan imgede düzelme olup olmadığına göre

süperpiksel algoritmalarının performansları incelenmektedir ve Şekil 3.3’de verilen blok diyagram ile gerçekleştirilmiştir.



Şekil 3.3. Filtreli imgeler için bölütleme blok diyagramı

Tez kapsamında gerçekleştirilen 3 aşama için gürültü tipleri, filtre türleri, kullanılan süperpiksel algoritmalarının özellikleri detaylı bir şekilde anlatılmaktadır.

3.1. Gürültü

Bilgisayarlı görü sistemlerinde, sinyal işleme, imge ve video işleme alanında veri toplama ve veri aktarımı sırasında gürültü ne yazık ki kaçınılmazdır. Gürültü imge veya ses üzerindeki istenmeyen değişimlere gürültü denilmektedir. Diğer bir deyişle gürültü imgedeki piksel değerindeki bozulmayı yani değişimini ifade etmektedir. İmge üzerinde gürültü uzamsal ve frekans özellikler olmak üzere iki kısımda incelenmektedir. Uzamsal gürültü özellikleri, gürültülü imge ile orijinal imge arasındaki korelasyon olup olmadığı durum ile açıklanmaktadır. Frekans gürültü özellikleri ise fourier doğrultusu yani elektromanyetik spektrumun frekans içeriğine karşılık gelmektedir. Örneğin, gürültünün fourier spektrumu sabit olduğunda, gürültü genellikle beyaz gürültü olarak adlandırılmaktadır (Gonzales,1992).

Gauss gürültüsü, rayleigh gürültüsü, gama gürültüsü, tuz&biber gürültüsü, bulanıklaştırma gibi birçok imge gürültüsü bulunmaktadır. Bu çalışma

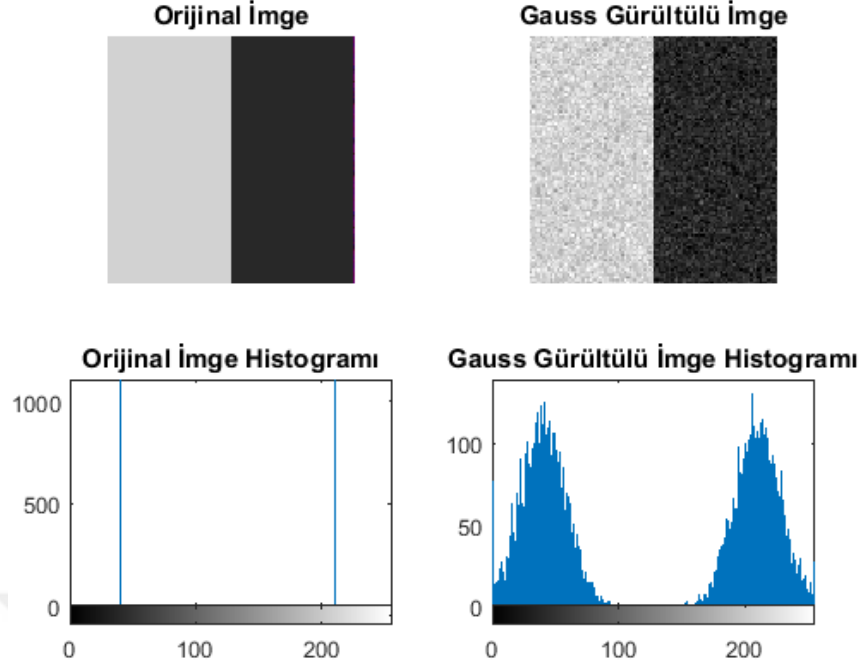
kapsamında tuz&biber gürültüsü, gauss gürültüsü ve bulanıklaştırma kullanılmıştır. Bu gürültüler aşağıda detaylı bir şekilde anlatılacaktır.

3.1.1. Gauss Gürültüsü

Gauss gürültüsü matematiksel kullanım nedeniyle uzamsal ve frekans düzlemlerinde sıklıkla kullanılmaktadır. Gauss gürültünün diğer bir adı normal gürültüdür. Gauss gürültüde piksel değerinin değişimi olasılık yoğunluk fonksiyonu (Probability Density Function, PDF) ile gerçekleştirilmektedir. z yoğunluk değerini, \bar{z} , z yoğunluk değerinin ortalama değerini ifade etmektedir. σ değeri standart sapmayı ve σ^2 değeri standart sapmanın karesi yani varyansı göstermektedir.

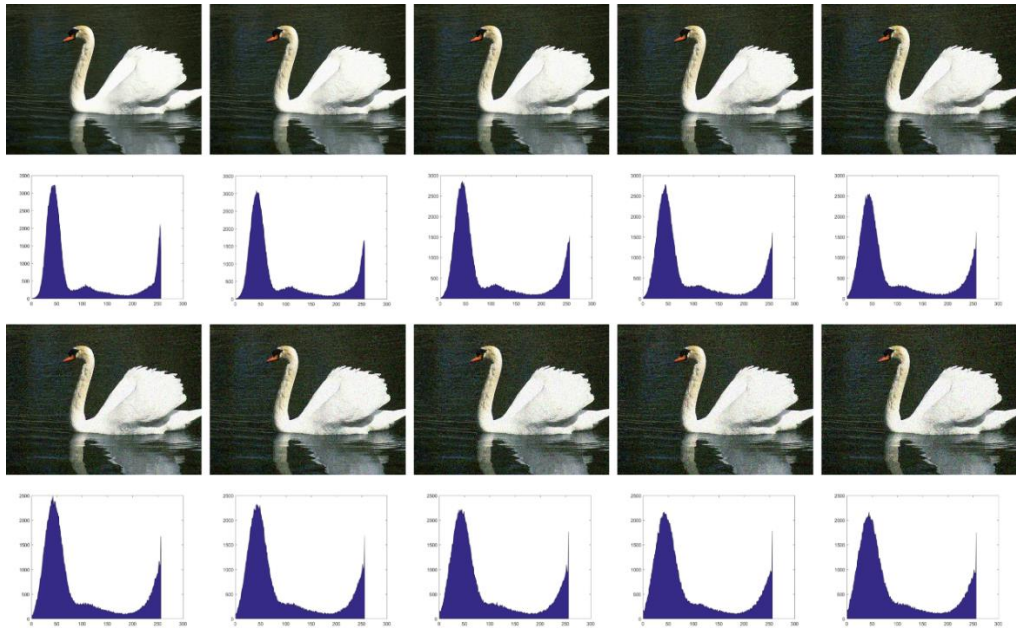
$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\bar{z})/2\sigma^2} \quad (3.1)$$

Şekil 3.4'te gauss gürültüsünün etkisi ile imge üzerindeki değişimi incelenmek amacıyla sentetik imgeye ortalama değeri 0, varyans değeri 0.01 değerinde gauss gürültüsü uygulanmıştır. Burada değişimi hem görsel olarak hem de histogram dağılımı ile piksel değerlerinin değişimleri görülmektedir. Orijinal imge iki yoğunluk değerine sahip 100×100 boyutlarındadır. Gauss gürültülü imgenin histogramı varyans değerine bağlı olarak yeni dağılımı göstermektedir. Orijinal imge iki yoğunluk değerindeyken gürültülü imge yoğunluk değerleri birbirine yaklaşan piksel yoğunluk değerleri gözlemlenmektedir.



Şekil 3.4. Sentetik orijinal imge ve gauss gürültülü imge ve histogram dağılımları

Bu tez çalışmasında gauss gürültüsü için 0 ortalama değeri ve 0.001, 0.002, ..., 0.01 varyans değeri kullanılarak test imgelerine 10 farklı seviyede gürültü uygulanmıştır. Şekil 3.5'te test imgesine uygulanan 10 farklı gauss gürültü seviyesi için hem imge hem de histogram değişimleri verilmektedir.



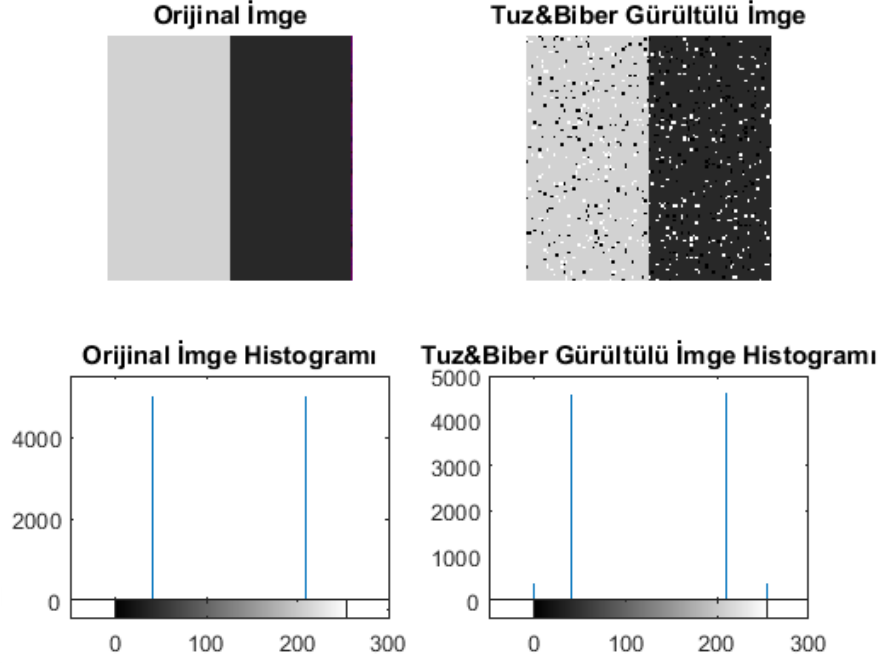
Şekil 3.5. Test imgesi için orijinal imge ve gauss gürültülü imge ve histogram dağılımları

3.1.2. Tuz&Biber Gürültüsü

Tuz&biber gürültüsü dürtü gürültüsü olarak da bilinmektedir. Tuz&biber gürültüsünün PDF değeri eşitlik 3.2'deki gibi verilmiştir. Burada $b > a$ ise b yoğunluğu imgede açık bir nokta olarak görülürken, a yoğunluğu ise koyu bir nokta olarak görülmektedir. P_a ve P_b değerlerinden biri sıfır ise tek kutuplu gürültü olarak adlandırılır. Her iki olasılık değeri de sıfırdan farklıysa tuz&biber gürültüsü oluşmaktadır.

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{diğer} \end{cases} \quad (3.2)$$

Gürültü dürtüleri negatif veya pozitif olabilir. İmgede dürtü bozukluğu genellikle uç noktalarda meydana gelmektedir. 8 bitlik bir imge için pozitif değer maksimum yani 255 değeri olmaktadır. Negatif değer ise minimum yani 0 değeridir. 8 bitlik bir imgede 255 beyazı, 0 siyahı gösterdiği için tuz ve bibere benzediği için bu isimlendirme yapılmıştır [Gonzales,(1992)]. Şekil 3.6'te %8 gürültü yoğunluğu olacak şekilde eklenmiştir. Tuz&biber gürültülü imgenin histogramı incelendiğinde 0 ve 255 değerlerinde rasgele piksel değerleri eklenmiştir.



Şekil 3.6. Sentetik orijinal imge ve tuz&biber gürültülü imge ve histogram dağılımları

Bu tez kapsamında tuz&biber gürültüsü için %4, %8, %12 ve %16 olacak şekilde 4 farklı gürültü yoğunluğu test imgelerine eklenmiştir. Şekil 3.7’de test imgesine uygulanan 4 farklı tuz&biber gürültüsünün imge ve histogram dağılımının değişimi verilmektedir.

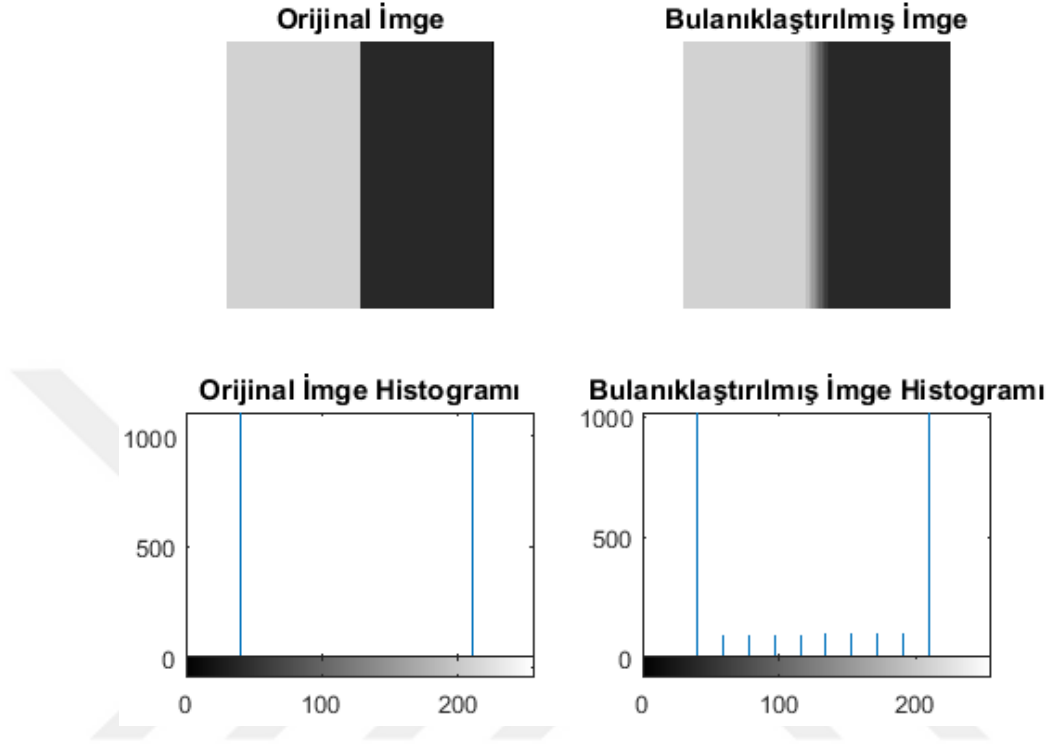


Şekil 3.7. Test imgesi için orijinal imge ve tuz&biber gürültülü imge ve histogram dağılımları

3.1.3. Bulanıklaştırma

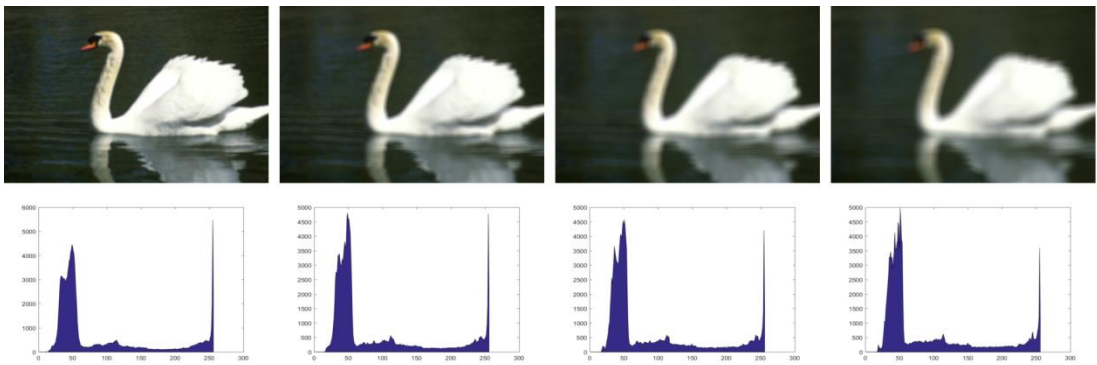
Bulanıklaştırma işlemi filtreleme işlemi gibi düşünülebilir. Burada imge üzerinde kayan pencere yöntemiyle pencerenin ortanca piksel değerlerinin güncellenmesiyle oluşmaktadır. Bulanıklaştırma işlemi yapılırken piksel

değerleri birbirine yaklaşmaktadır. Burada pencere boyutu ne kadar büyük olursa bulanıklaştırma da o kadar fazla olmaktadır. Ayrıca bulanıklaştırma işlemi ile keskin sınır geçişler zayıflamaktadır. Şekil 3.8'de 9×9 pencere boyutu kullanılmıştır.



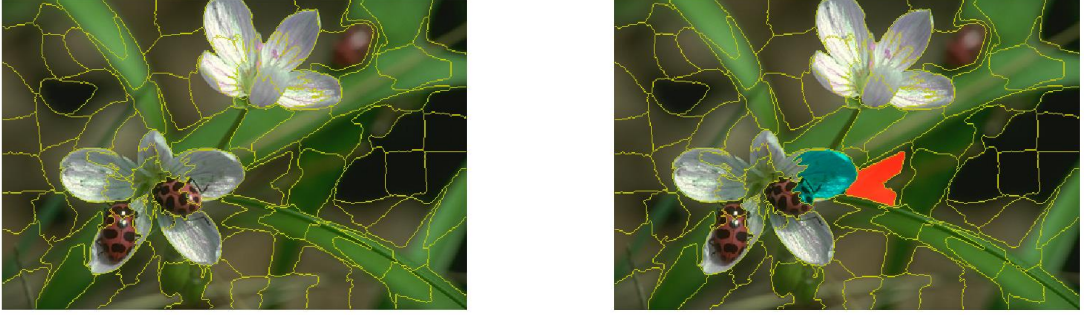
Şekil 3.8. Sentetik orijinal imge ve bulanıklaştırılmış imge ve histogram dağılımları

Bu tez kapsamında bulanıklaştırma için 5×5 , 9×9 , 13×13 ve 17×17 pencere boyutlarında 4 farklı pencere ile Şekil 3.9'da verilen test imgelerinden biri için bulanıklaştırma işlemi yapılmıştır.



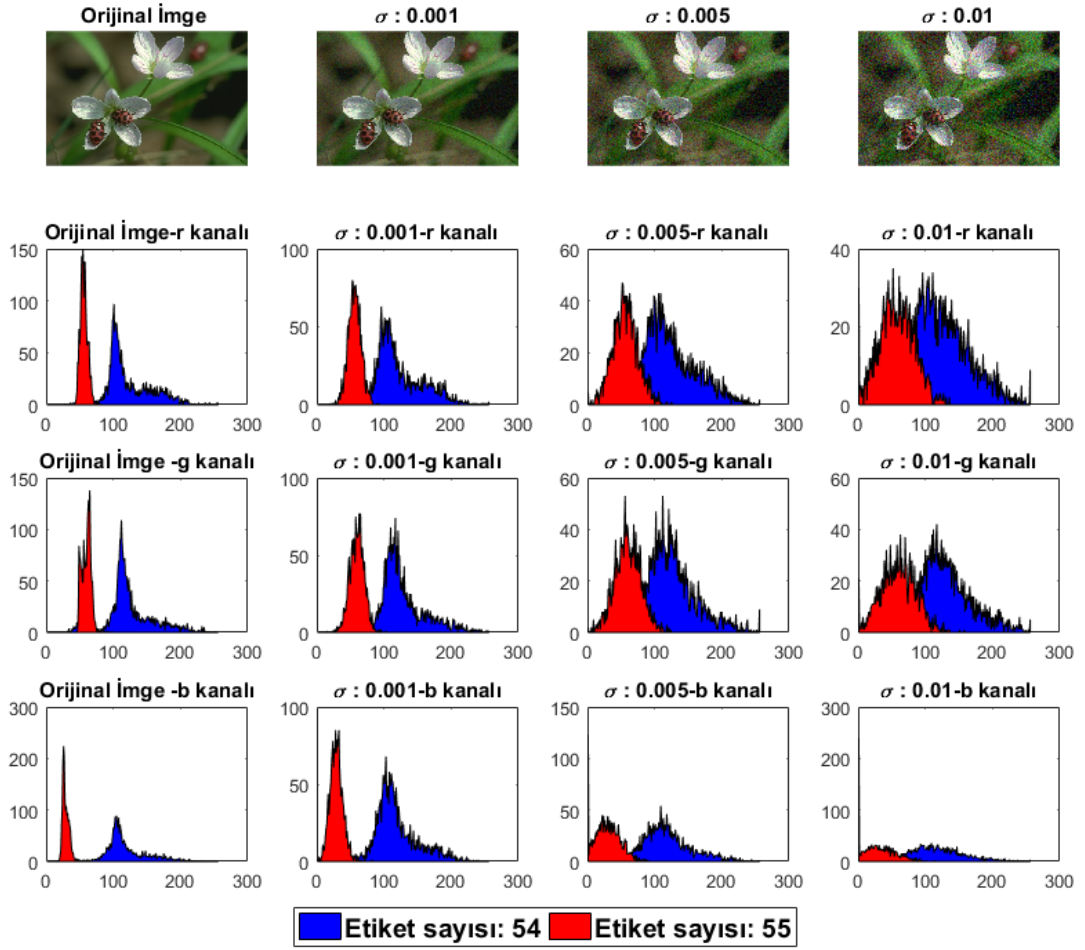
Şekil 3.9. Test imgesi için orijinal imge ve bulanıklaştırılmış imge ve histogram dağılımları

Süperpiksel algoritmaları sonucunda oluşan süperpiksellerin gürültüye bağlı olarak davranışının değişimini incelemek için SLIC algoritması ile bir imge 100 süperpiksele bölütlenmiştir. Şekil 3.10'da SLIC algoritmasının 100 süperpiksel için gösterimi verilmiştir. Ayrıca mavi ve kırmızı bölge iki komşu süperpikseli ifade etmektedir ve sırasıyla etiket sayısı 54 ve 55'tir.



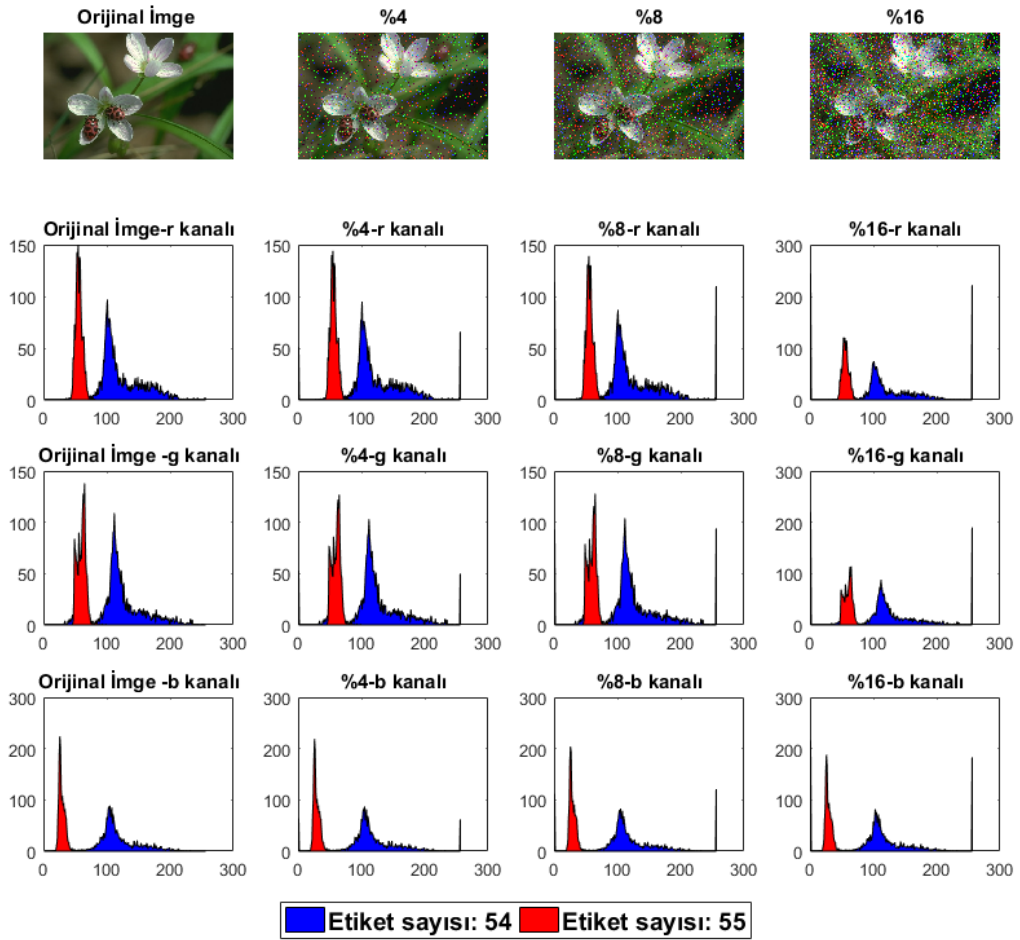
Şekil 3.10. SLIC algoritması sonucunda iki komşu süperpikselin gösterimi

Şekil 3.11'de gauss gürültüsüne bağlı olarak iki komşu süperpikselin piksel değerlerinin değişimini incelemek için histogram dağılımları verilmiştir. Burada orijinal imge ve 3 farklı gauss gürültü seviyesi uygulanarak piksel yoğunluk değişimleri her bir renk kanalı yani r,g,b kanalı için ayrı ayrı incelenmiştir. Orijinal imgenin histogram dağılımları iki ayrı yoğunluk tepelerine sahipken gürültü seviyesinin artmasıyla yoğunluk tepeleri birbirine yaklaşmaktadır hatta kesişimde bulunan piksel yoğunluk değerlerinin sayısı artmaktadır. Şekil 3.10'da verilen mavi bölge histogram dağılımında da mavi bölgeyi yani 54. etiketi tanımlamaktadır. Aynı şekilde kırmızı bölge histogram dağılımında da kırmızıyı bölgeyi yani 55. etiketi ifade etmektedir.



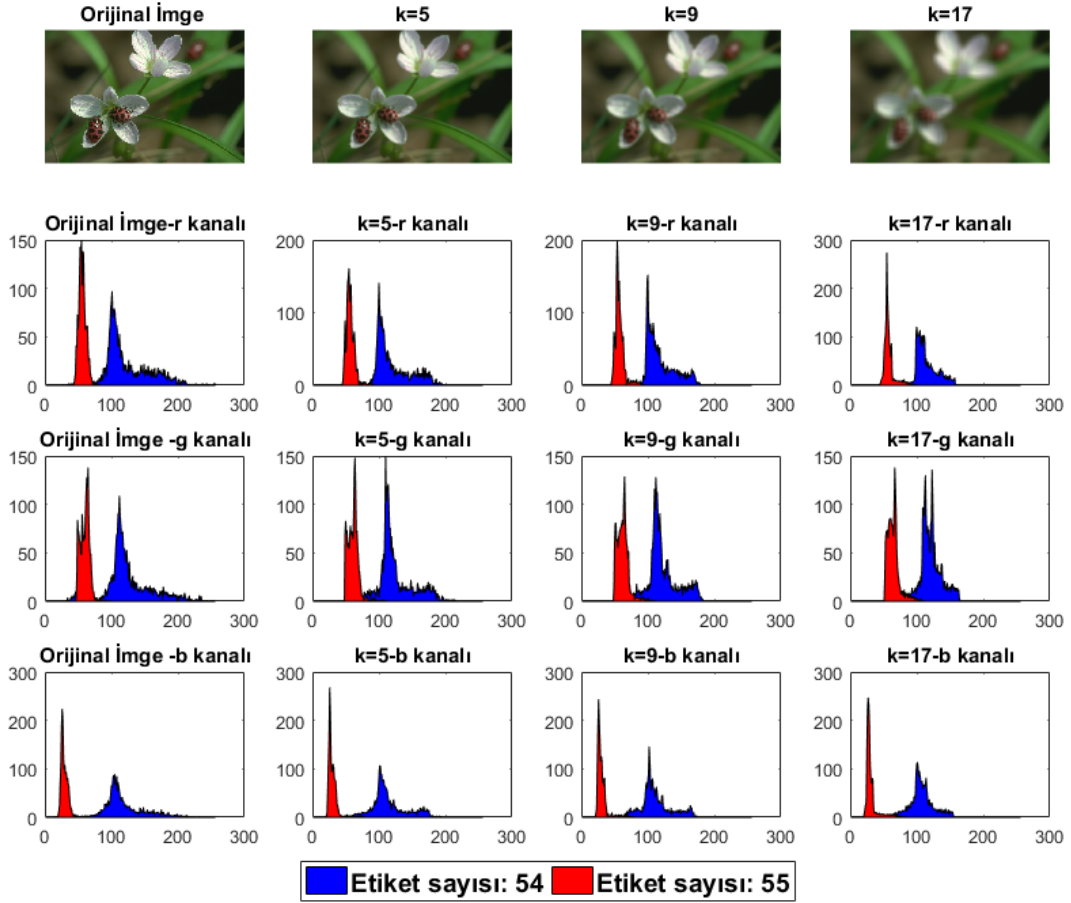
Şekil 3.11. İki komşu süperpikselin gauss gürültüsüne bağlı olarak histogram değişimlerinin gösterimi

Şekil 3. 12’de tuz&biber gürültüsüne bağlı olarak iki komşu süperpikselin piksel değerlerinin değişimini incelemek için histogram dağılımları verilmiştir. Burada orijinal imge ve 3 farklı tuz&biber gürültü seviyesi uygulanarak piksel yoğunluk değişimleri her bir renk kanalı yani r,g,b kanalı için ayrı ayrı incelenmiştir. Tuz&biber gürültüsü eklenen imgelerde tuz değeri 255 ve biber değeri 0 olan iki piksel değerinde eklenen yüzde kadar artış olmaktadır. Burada iki süperpikselin karışması söz konusu değildir. Sadece 0 ve 255 değerleri artacağından diğer piksel değerinde azalma meydana gelmiştir yani sönümlenme söz konusudur.



Şekil 3.12. İki komşu süperpikselin tuz&biber gürültüsüne bağlı olarak histogram değişimlerinin gösterimi

Şekil 3.13'te bulanıklaştırma etkisine bağlı olarak iki komşu süperpikselin piksel değerlerinin değişimini incelemek için histogram dağılımları verilmiştir. Burada orijinal imge ve 3 farklı bulanıklaştırma seviyesi uygulanarak piksel yoğunluk değişimleri her bir renk kanalı yani r,g,b kanalı için ayrı ayrı incelenmiştir. Bulanıklaştırma ile belirlenen pencere boyutlarındaki piksellerin değerleri birbirine yaklaşmaktadır. Böylece süperpiksellerin varyans değeri düşmektedir. Bunun sonucunda bulanıklaştırma etkisiyle oluşturulan yeni imgeler süperpiksellere ayrıldığında daha homojen süperpikseller elde edilmektedir.



Şekil 3.13. İki komşu süperpikselin bulanıklaştırma etkisine bağlı olarak histogram değişimlerinin gösterimi

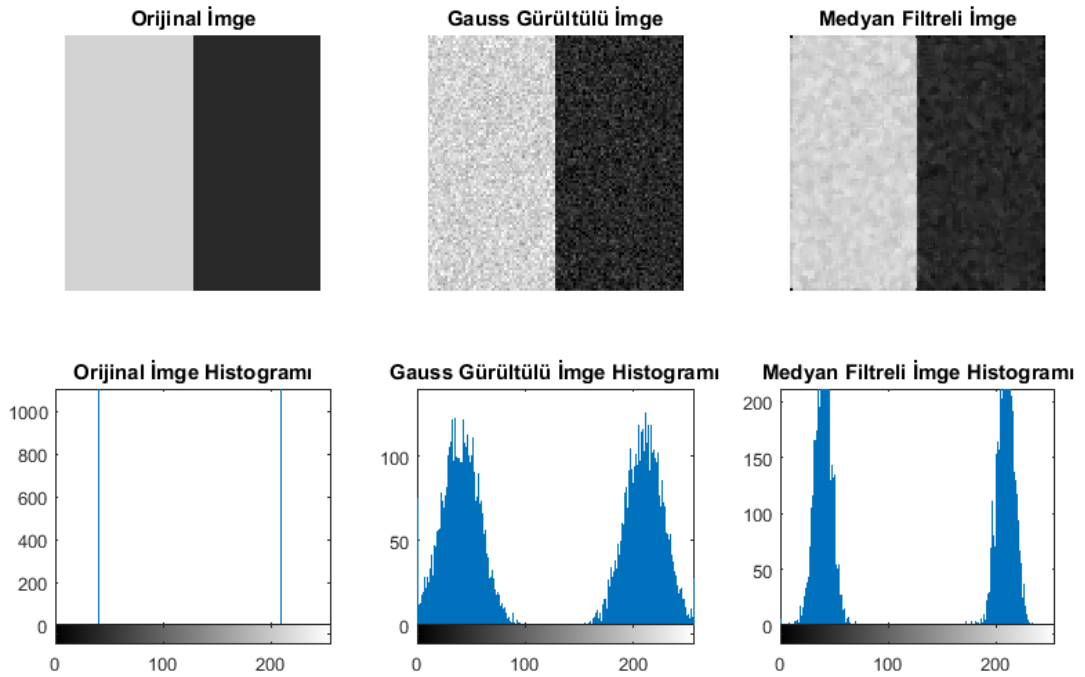
3.2. Filtre

İmge işleme uygulamalarında filtreler imge üzerindeki ayrıntıların daha belirgin hale gelmesi için kullanılmaktadır. Bir nesnenin kenarının keskinleştirilmesi, nesne kenarının yakalanması ve gürültülü imgeler için gürültü giderimi gibi birçok uygulamada filtreler aktif rol almaktadır. Literatürde filtreleme iki ana başlık altında toplanmıştır. Bunlardan ilki uzamsal filtreleme ve uzamsal filtreleme işlem yaparken imge üzerinde hiçbir dönüşüm yapmadan imgeye doğrudan uygulanmaktadır. İkinci filtreleme ise frekans bölgesinde olup imge üzerinde işlem yapılırken fourier dönüşümü ile frekans alanına dönüştürülmektedir. İmge üzerinde filtreleme işlemi yapıldıktan sonra tekrar ters fourier dönüşümü işlemi uygulanarak filtrelenmiş imge elde edilmektedir. Bu tez kapsamında filtreler gürültü giderimi için kullanılmıştır. Ayrıca gürültü

giderimi için uzamsal filtrelerden medyan filtre ve frekans bölgesindeki filtrelerden wiener filtresi tercih edilmiştir.

3.2.1. Medyan Filtre

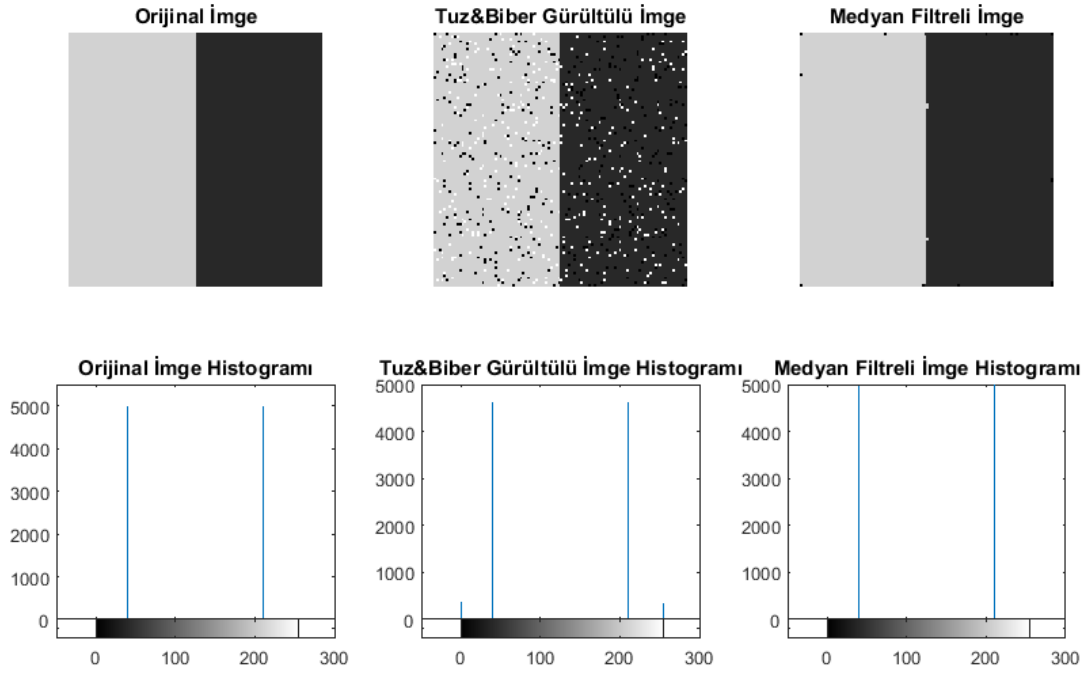
Medyan filtre alçak geçiren filtrenin özel bir türüdür ve gürültü giderimi için sıklıkla kullanılmaktadır. Medyan filtreleme için konvolüsyon kullanımı gerektirmez. Filtreleme işlemi yapılırken pencere boyutu seçilerek bu penceredeki pikseller küçükten büyüğe doğru sıralanmaktadır. Daha sonra ortanca değer bulunarak ortadaki piksel değeri güncellenmektedir. Şekil 3.14'te orijinal imge, gauss gürültülü imge ve gauss gürültülü imgeye medyan filtre uygulanan imge ve histogram dağılımları verilmiştir. Burada medyan filtre ile gürültülü imgeye göre orijinal imgeye daha da yaklaştığı histogram dağılımından görülmektedir.



Şekil 3.14. Sentetik orijinal imge, gauss gürültülü imge ve medyan filtreli imge ve histogram dağılımları

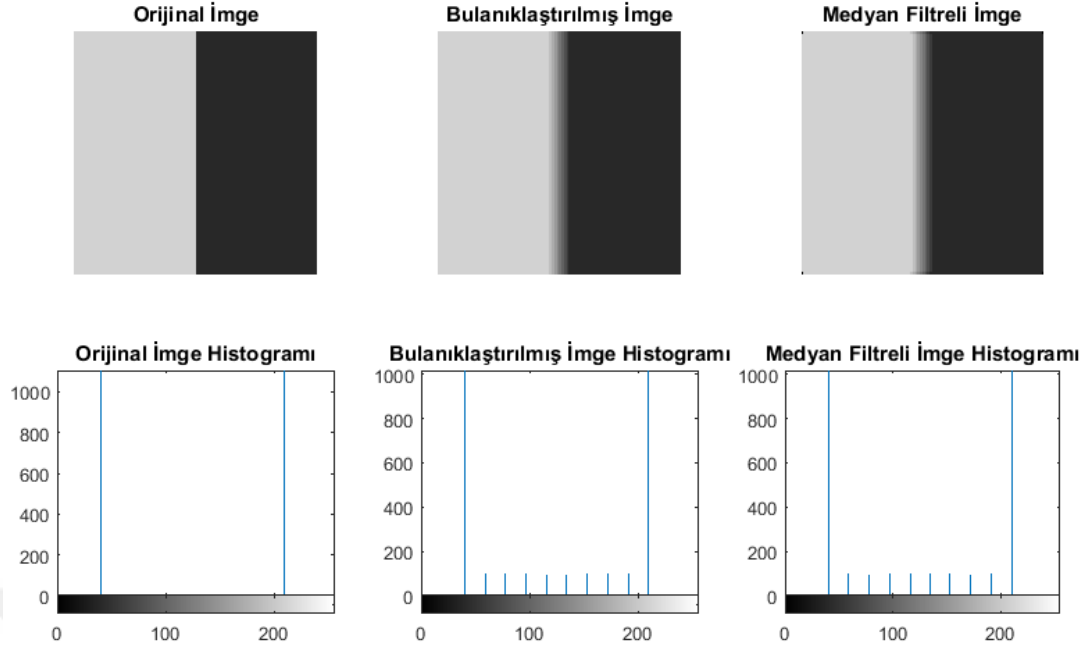
Şekil 3.15'te orijinal imge, tuz&biber gürültülü imge ve medyan filtreli imge ve histogram dağılımları verilmiştir. Burada medyan filtrenin en iyi sonuç verdiği tuz&biber gürültülü imge olduğu medyan filtreli imge ve histogram dağılımından

açıkça belli olmaktadır. Çünkü tuz&biber gürültüsü yüzde yoğunluğu ile imgeye eklenmektedir ve medyan filtrede pencere boyutu içerisindeki pikselleri küçükten büyüğe doğru sıraladığı için gürültülü piksel değeri ortadaki piksel değeri ile güncellenmektedir. Böylece minimum değer 0, maksimum değer 255 olduğu durumda ortada bulunan değer imgenin orijinal değerine daha yakın olduğundan medyan filtre tuz&biber gürültü giderimi için oldukça başarılıdır.



Şekil 3.15. Sentetik orijinal imge, tuz&biber gürültülü imge ve medyan filtreli imge ve histogram dağılımları

Şekil 3.16'da orijinal imge, bulanıklaştırılmış imge ve medyan filtreli imge ve histogram dağılımları verilmiştir. Bulanıklaştırılmış imgeye uygulanan medyan filtre imgeyi iyileştirmede çok başarılı olmadığı gözlenmektedir.



Şekil 3.16. Sentetik orijinal imge, bulanıklaştırılmış imge ve medyan filtreli imge ve histogram dağılımları

3.2.2. Wiener Filtre

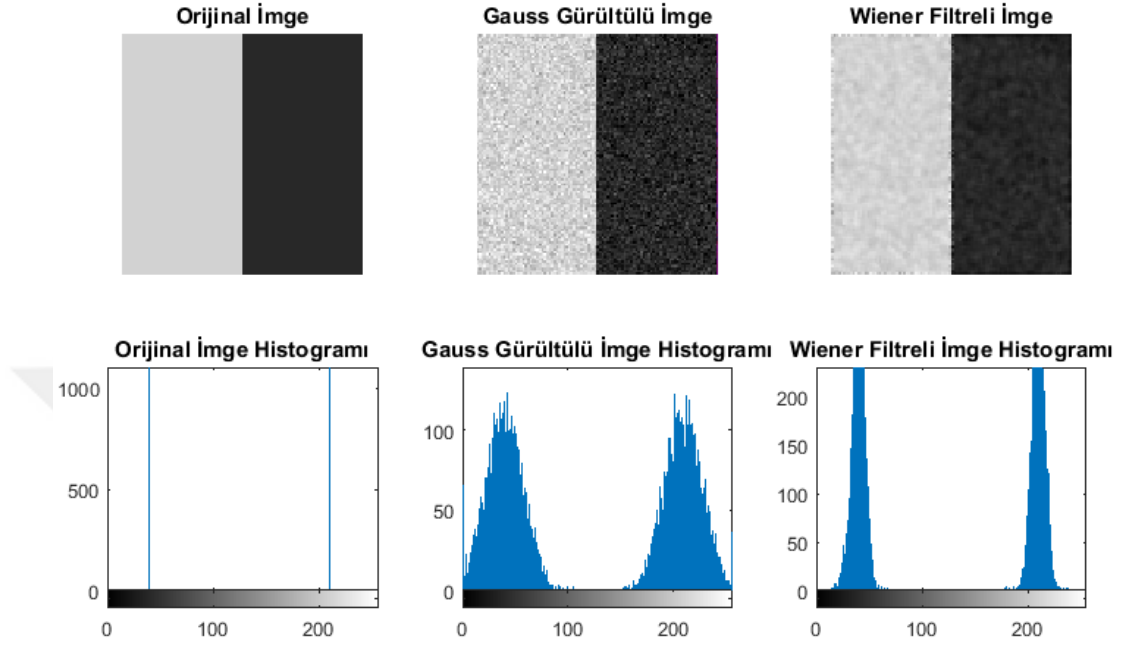
En küçük ortalama karesel hata filtresi olarak da bilinen wiener filtresi frekans bölgesinde çalışmaktadır. Wiener filtre istatistiksel bir yaklaşım ile hata karelerinin ortalamasını en aza indirmeye çalışarak filtreli imgeyi oluşturmaktadır. Wiener filtre denklem 3.3'teki gibi hesaplanarak imge üzerinde güncelleme yapılmaktadır.

$$\hat{F}(u, v) = \left[\frac{H^*(u, v)S_f(u, v)}{S_f(u, v)|H(u, v)|^2 + S_\eta(u, v)} \right] G(u, v) \quad (3.3)$$

$H(u, v)$ bozunma fonksiyonunu, $G(u, v)$ bozulmuş imgeyi temsil etmektedir. $S_f(u, v)$ ve $S_\eta(u, v)$ orijinal imgenin güç spektrumu ve gürültünün güç spektrumunu sırasıyla ifade etmektedir.

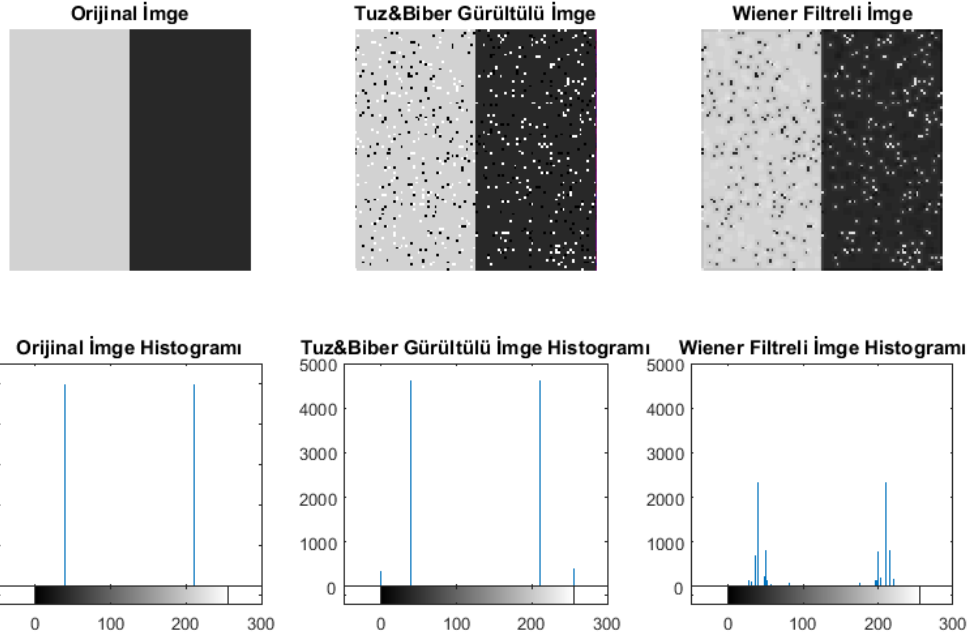
Şekil 3.17'de orijinal imge, gauss gürültülü imge ve wiener filtreli imge ve histogram dağılımları verilmiştir. Wiener filtrenin medyan filtreye göre gauss

gürültüsünde daha başarılı olduğu gözlenmektedir. Medyan filtrede daha büyük standart sapma değerine sahipken wiener filtrede daha küçük standart sapmaya sahiptir. Bu sonuçla gauss gürültü giderimi için wiener filtrenin daha iyi sonuçlandığı gözlenmektedir.

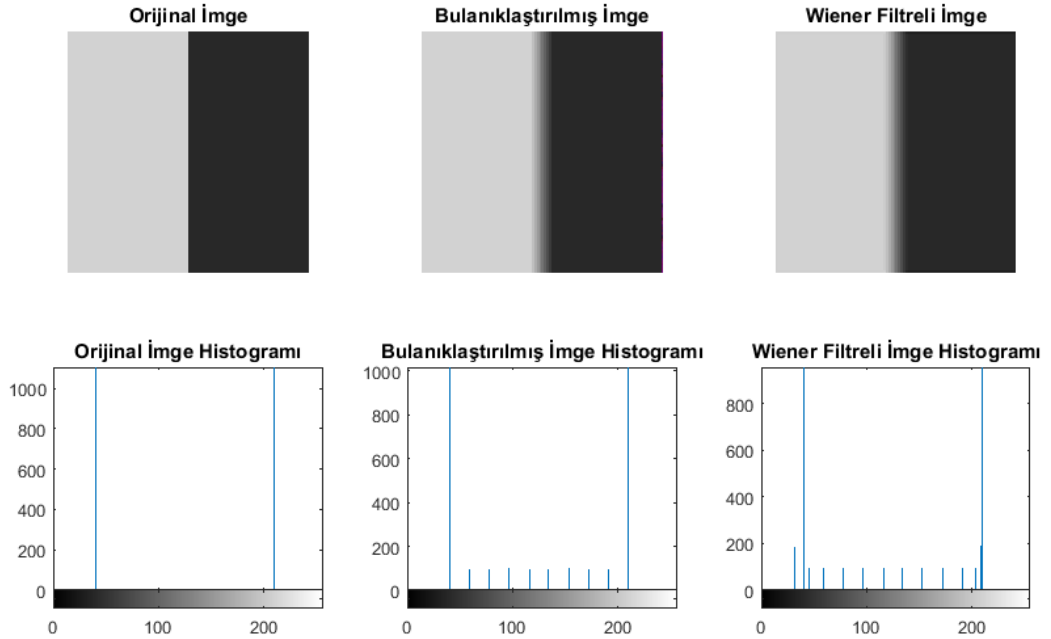


Şekil 3.17. Sentetik orijinal imge, gauss gürültülü imge ve wiener filtreli imge ve histogram dağılımları

Şekil 3.18'de orijinal imge, tuz&biber gürültülü imge ve wiener filtreli imge ve histogram dağılımları verilmiştir. Burada wiener filtrenin tuz&biber gürültüsü üzerinde gürültü gideriminde pek başarılı olmadığı görülmektedir. Şekil 3.19'da orijinal imge, bulanıklaştırılmış imge ve wiener filtreli imge verilmiştir. Bulanıklaştırılmış imgeyi iyileştirmek için medyan filtrede olduğu gibi wiener filtrede pek başarılı değildir yani sonuçların değişmediği histogram dağılımında görülmektedir.



Şekil 3.18. Sentetik orijinal imge, tuz&biber gürültülü imge ve wiener filtreli imge ve histogram dağılımları



Şekil 3.19. Sentetik orijinal imge, bulanıklaştırılmış imge ve wiener filtreli imge ve histogram dağılımları

3.3. Süperpiksel Algoritmaları

Literatürde en çok kullanılan süperpiksel algoritmalarından SLIC (2012), LSC (2016), RPS (2014), ERC (2014) ve TP (2009) algoritmalarının orijinal ve

gürültülü imgeler üzerindeki performansları incelenmiştir. Bu algoritmaların tercih edilme nedenlerinin başında bilgisayarlı görü sistemlerinde en çok kullanılan algoritmalar olmasıdır. Bir başka sebebi ise farklı yöntemlerden seçilen bu algoritmaların doğruluğu karşılaştırılabilir derecede olmasıdır. Aşağıda bu algoritmalar detaylı bir şekilde anlatılacaktır.

3.3.1. Basit Doğrusal Yinelemeli Kümeleme Algoritması (SLIC)

SLIC var olan çalışmalardan daha hızlı, hafıza verimliliği yüksek, sınır uyumluluğu oldukça iyi bir algoritmadır. Kümeleme tabanlı süperpiksel algoritmaları arasında en çok kullanılan algoritmaların başında gelmektedir. Algoritma hem gri seviyeli imgelere hem de renkli imgelere uygulanabilmektedir. Renkli imgelere uygulanırken CIELAB renk uzayına dönüştürülerek kullanılmaktadır. Uluslararası Aydınlatma Komisyonu(Commission Internationale de l'Eclairage, CIE) tarafından insan gözü algılamasına en yakın olarak tasarlanan renk uzayıdır. Burada L açıklık ya da parlaklığı ifade etmektedir ve değeri $[0,100]$ arasındadır. $L=0$ siyahı, $L=100$ beyazı ifade etmektedir. a kırmızı/yeşil koordinatına karşılık gelmektedir. a^+ kırmızıyı ve a^- yeşili temsil etmektedir. b ise sarı/mavi koordinatını göstermektedir. Ayrıca b^+ sarıyı b^- ise maviyi belirtmektedir. Algoritmanın başlayabilmesi için tek parametreye ihtiyaç vardır. Bu parametre istenilen sayıdaki süperpiksel sayısını gösteren k parametresidir. Kümeleme tabanlı süperpiksel algoritmalarının ortak özelliği olan başlangıç kümelerinin oluşturulmasıyla başlamaktadır. İlk olarak oluşturulan başlangıç kümeleri için merkez noktaları 3×3 pencere boyutundaki noktalar arasındaki en küçük gradyan bilgisine göre belirlenmektedir. Başlangıç kümelerinin merkezlerini içeren $C_i = [l_i, a_i, b_i, x_i, y_i]^T$ beş boyutlu piksel vektörü oluşturulmaktadır. Burada l, a, b renk bilgilerini ve x, y konum bilgilerini temsil etmektedir. Daha sonra imgeye ait tüm piksellerin hangi süperpiksele ait olduğu atanır. Bu atama işlemi temel olarak k -ortalama kümeleme algoritmasının mantığına dayanmaktadır. Burada her bir süperpikselin merkez noktası ile imgeye ait piksellerin arasındaki mesafeye bakılarak o pikselin en yakın olduğu süperpiksel belirlenir. Mesafe hem renk bilgilerini hem de konum bilgilerini içeren bir Öklid mesafesidir. Bir pikselin etiketine mesafe değeri en küçük olan süperpikselin etiket değeri atanır. Atama

gerçekleştikten sonra her bir süperpiksel merkezi, süperpiksel içerisindeki piksellerin ortalaması alınarak güncellenmektedir. Bu güncelleme işlemi hata değerinin belirlenen eşik değerince yinelenmektedir. Hata değeri yeni merkezlerle eski merkez arasında öklid mesafesinin hesaplanmasıyla elde edilmektedir. Ancak tüm piksellerin tek tek kontrol edilmesi işlem yükünü arttırdığından SLIC algoritmasında işlem yükünün azaltılması için k-ortalamlar kümeleme algoritmasında yeniden düzenlenmiştir ve imgenin tüm uzayının kontrol edilmesi yerine $2S \times 2S$ arama uzayı kullanılmaktadır. Burada S başlangıç adımında belirlenen süperpikselin kenar uzunluğudur ve $S = \sqrt{N/k}$ formülü ile hesaplanmaktadır. N imgenin tüm piksel sayısı, k ise süperpiksel sayısını gösteren algoritma için gerekli tek parametredir.

$$\begin{aligned}
 d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\
 d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\
 D &= \sqrt{d_c^2 + m^2 (d_s / S)^2}
 \end{aligned} \tag{3.4}$$

Denklem 3.4' de d_c ve d_s oluşturulan süperpikseller ile imgeye ait piksellerin sırasıyla renk ve konum değerleri arasındaki mesafeyi göstermektedir. D ise etiket atama işleminde kullanılan mesafe ölçüsünü göstermektedir. Burada m , renk ve konum değerleri arasındaki ağırlık ölçüsüdür ve [1,40] aralığındadır. Şekil 3.20'de 200 süperpiksel sayısı için sırasıyla m 1, 5, 10, 25, 35 ve 40 değerinin 6 farklı performansı verilmiştir. m değeri 40'a yaklaştıkça oluşan süperpiksellerin şekillerinin daha düzgün bir yapıda, 1'e yaklaştıkça süperpiksellerin daha homojen yapıda oluşmaktadır. Bu çalışmada m değeri 10 olarak alınmıştır. Ayrıca oluşan süperpikseller içerisinde küçük küçük başka etiketlere ait pikseller olduğu için işlem sonrası gerçekleştirilmiştir. Böylece oluşan süperpikseller aynı etikete sahip tek grup oluşmaktadır. SLIC algoritmasının sözde kodu aşağıda verilmiştir.



Şekil 3.20. SLIC algoritmasının m değerine göre 200 süperpiksel için sonuçları

% Ön Değer Atama

S uzunluğuna sahip düzenli örgü yapıları oluşturulur.

Başlangıç küme merkezleri belirlenir.

Küme merkezleri en küçük gradyan yönünde güncellenir.

Her bir pikselin etiketi $l(i) = -1$ olarak atanır.

Her bir pikselin mesafesi $d(i) = \infty$ olarak atanır.

While ($E >$ eşik değeri)

%Atama

for her bir süperpiksel için

for merkez etrafında $2S \times 2S$ alanındaki pikseller için

D= süperpiksel ile pikseller arasında Öklid mesafesi hesaplanır.

if $D < d(i)$

$d(i) = D$

$l(i) = k$

end

end

end

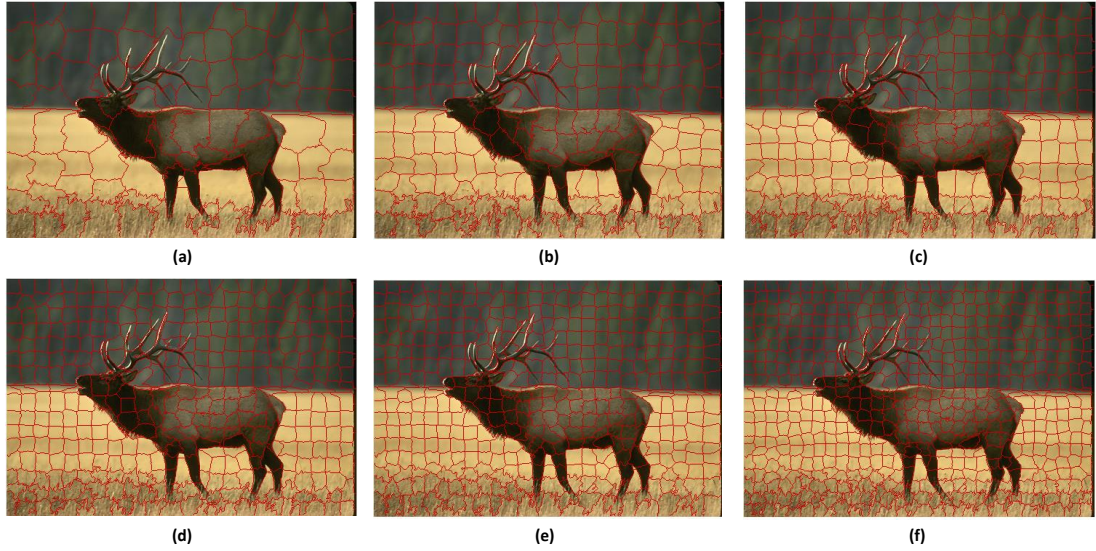
%Güncelleme

Yeni küme merkezleri hesaplanır.

Artık hata değeri E hesaplanır.

end

Şekil 3.21'de 100-600 olmak üzere 6 farklı süperpiksel sayısı için SLIC süperpiksel algoritmasının performansı verilmiştir. Süperpiksel sayısı arttıkça sınır uyumluluğunun da arttığı görülmektedir.



Şekil 3.21. SLIC algoritmasının farklı süperpiksel sayıları için süperpiksel performansları

3.3.2. Doğrusal Spektral Kümeleme Algoritması (LSC)

Doğrusal spektral kümeleme olarak adlandırılan ve düşük hesaplama maliyetleri ile doğal imgeler için hem yüksek sınır uyumu hem de görsel olarak kompaktlığa sahip süperpikseller üretebilen bir süperpiksel bölütleme algoritmasıdır. İmge pikselleri arasında hem alan yakınlığı hem de renk benzerliğini ölçen mesafe metriği kullanılarak imge bölütlemenin normalize kesme tabanlı (Ncut) formülasyonuna uyarlanmıştır. Normalize kesme çalışmasında olduğu gibi direk olarak öz tabanlı algoritma kullanmak yerine piksel değerleri yüksek boyutlu öznitelik uzayına haritalandırılmıştır. Bu özellik uzayında her bir noktanın uygun bir şekilde ağırlıklandırılmasıyla, ağırlıklandırılmış k-ortalamlar (kmeans) ile normalleştirilmiş kesimlerin amaç fonksiyonları aynı optimum noktaları temsil etmektedir. Böylece önerilen öznitelik uzayında k ortalamlar kümeleme algoritması yinelemeli olarak uygulanarak normalize kesme algoritmasının maliyet fonksiyonu optimize edilmektedir. Ağırlıklı k-ortalamlar yönteminin amaç fonksiyonu

$$F_{km} = \sum_{k=1}^K \sum_{p \in \pi_k} w(p) \|\phi(p) - m_k\|^2 \quad (3.5)$$

olarak tanımlanmaktadır. Her veri noktası p , bir ağırlık $w(p)$ ile atanmaktadır. π_k k . kümeyi, K küme sayısını temsil etmektedir. $\phi(p)$ doğrusallığı arttırmak

için veri noktalarını yüksek boyutlu bir özellik alanına eşleyen fonksiyonudur. m_k küme merkezlerini ifade etmektedir ve denklem 3.6'daki gibi hesaplanmaktadır.

$$m_k = \frac{\sum_{q \in \pi_k} w(q)\phi(q)}{\sum_{q \in \pi_k} w(q)} \quad (3.6)$$

K-yol normalize kesme yönteminin amaç fonksiyonu

$$F_{Ncut} = \frac{1}{K} \sum_{k=1}^K \frac{\sum_{p \in \pi_k} \sum_{q \in \pi_k} W(p, q)}{\sum_{p \in \pi_k} \sum_{q \in V} W(p, q)} \quad (3.7)$$

olarak tanımlanmaktadır. $W(p, q)$ iki veri noktası p ve q arasındaki benzerliği hesaplamaktadır. Burada iki veri noktası p ve q beş boyutlu renk ve konumlarını içeren piksel vektörleridir. Bu algorithmada her piksel on boyutlu bir öznitelik uzayının içindeki bir noktaya eşlenmektedir.

$$\begin{aligned} \phi(p) = \frac{1}{w(p)} & (C_c \cos \frac{\pi}{2} l_p, C_c \sin \frac{\pi}{2} l_p, 2.55C_c \cos \frac{\pi}{2} \alpha_p, \\ & 2.55C_c \sin \frac{\pi}{2} \alpha_p, 2.55C_c \cos \frac{\pi}{2} \beta_p, 2.55C_c \sin \frac{\pi}{2} \beta_p, \\ & 2.55C_s \cos \frac{\pi}{2} x_p, 2.55C_s \sin \frac{\pi}{2} x_p, 2.55C_s \cos \frac{\pi}{2} y_p, 2.55C_s \sin \frac{\pi}{2} y_p) \end{aligned} \quad (3.8)$$

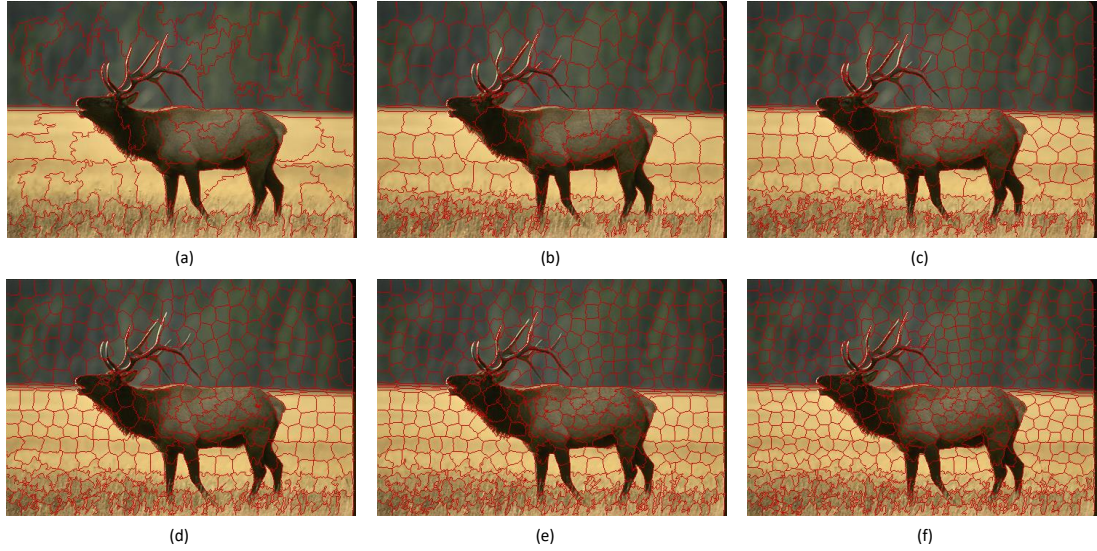
$$w(p) = \sum_{q \in V} W(p, q) = w(p)\phi(p) \cdot \sum_{q \in V} w(q)\phi(q)$$

Süperpikseller on boyutlu uzayda ağırlıklı k-ortalamlar yöntemiyle vektörlerin iteratif olarak gruplanmasıyla meydana gelmektedir. LSC algoritmasında da SLIC algoritmasında olduğu gibi $2S \times 2S$ alan içerisinde güncellenmektedir. Her iterasyon sonucunda süperpiksellerin ağırlıklı ortalamaları ve merkezleri güncellenmektedir. Ağırlık ortalamalar yakınsama işlemi gerçekleşinceye kadar devam etmektedir. Algoritma sonunda küçük süperpikseller oluşursa onlar birleştirilerek büyük süperpikseller oluşturulur. Bu algoritma için sözde kod

aşağıda verilmektedir. Algoritmada yer alan τ arama alanının katsayısı olup $\tau = 2$ olarak alınmıştır.

-
- 1: Her bir piksel CIELAB renk uzayına dönüştürülerek $p = (l_p, \alpha_p, \beta_p, x_p, y_p)$ öznelik uzayında 10 boyutlu vektöre haritalanır.
 - 2: K tane merkez belirlenir, imge sabit yatay ve dikey aralıklarla bölünür.
 - 3: 3×3 pencere boyutunda minimum gradyan hesaplanarak merkez konumları güncellenir.
 - 4: Tohumlara ait olan her bir kümenin merkezi c_k ve ağırlıklı ortalama m_k hesaplanır.
 - 5: Her p noktası için $L(p) = 0$ başlangıç etiketleri oluşturulur.
 - 6: Her p noktası için $D(p) = \infty$ başlangıç uzaklıkları oluşturulur.
 - 7: **repeat**
 - 8: for her kümenin merkezi c_k ve ağırlıklı ortalama m_k için
 - 9: for her p noktasının küme merkezinin $\tau v_x \times \tau v_y$ pencere boyutu için
 - 10: $D = \phi(p)$ ile ağırlıklı ortalama arasında Öklid mesafe hesaplama
 - 11: if $D < d(p)$
 - 12: $d(p) = D$
 - 13: $L(p) = k$
 - 14: end if
 - 15: end for
 - 16: end for
 - 17: Her küme için küme merkezi ve ağırlıklı ortalama güncellenir.
 - 18: **until** K kümenin ağırlıklı ortalamaya yakınsama gerçekleşene kadar
 - 19: Küçük süperpikseller komşuları ile birleştirilir.
-

Şekil 3.22’de 100-600 olmak üzere 6 farklı süperpiksel sayısı için LSC süperpiksel algoritmasının performansı verilmiştir. Süperpiksel sayısı arttıkça sınır uyumluluğunun da arttığı görülmektedir.



Şekil 3.22. LSC algoritmasının farklı süperpiksel sayıları için süperpiksel performansları

3.3.3. Düzenliliği Korunmuş Süperpiksel Algoritması (RPS)

Düzenliliği korunmuş süperpiksel (Regularity Preserved Superpixels, RPS) algoritmasında süperpiksel yapılarının düzenli olması amaçlanmaktadır. Düzenli süperpiksel kavramını değerlendirmek için iki kritere bakılmaktadır. Kriterlerden ilki, süperpikseller piksellerin bilgisini temsil etmektedirler ve piksellerden oluşan bölgelerin birbirleriyle uyumlu olması gerekmektedir. İkinci kriter ise piksellerin yapı, homojenlik, mekânsal topolojisi gibi kendine ait özellikleri koruması gerekmektedir. Algoritma genel anlamda üç adımdan oluşmaktadır. İlk adım bağlantı noktalarının bulunmasıdır. N süperpiksel sayısını ifade etmektedir. İmgenin genişliği ve yüksekliği sırasıyla H ve W olarak gösterilmektedir. Ayrıca bağlantı noktalarının yatay ve dikey boyutları da N_h ve N_w olarak ifade edilmektedir.

$$\begin{cases} W / N_w \approx H / N_h \\ N = N_w N_h \end{cases} \Rightarrow \begin{cases} N_h = \lfloor \sqrt{NH/W} \rfloor \\ N_w = \lfloor N / N_h \rfloor \end{cases} \quad (3.9)$$

Algoritmanın ikinci adımı ise bağlantı noktalarının yer değiştirme işlemidir. Bağlantı noktaları sınır haritasının maksimal genliğine göre optimal pozisyona hareket ettirilir. Komşu bağlantı noktalarıyla çarpışmayı sınırlandırmak için bölgesel arama uzayı belirlenir. Her bir bağlantı noktasının optimal pozisyonda

yer deęiřtirmesi iin iki kısıt vardır. Kısıtlardan biri süperpiksellerin baęlantı noktaları nesne sınırları üzerinde olmasıdır ve dięer kısıt ise süperpiksellerin konumsal yapılarını koruması gerekmektedir yani başlangıtaki ızgara yapıya yakın olması gerekmektedir. Denklem 3.10' da p_i aday pikseli, $f_b(p_i)$ pikselin sınır büyüklüğünü ve p_0 başlangı pikseli ile p_i aday pikseli arasında öklid uzaklığını hesaplamaktadır. Bu eřitlik ile baęlantı noktalarının yer deęiřtirilmesi saęlanmaktadır.

$$\hat{p} = \arg \max_{p_i \in \mathfrak{R}} [f_b(p_i) \cdot f_d(p_i, p_0)] \quad (3.10)$$

Algoritmanın adımı ise baęlantı noktalarının birleřtirilmesi adımıdır. Burada komřu baęlantı noktaları dikey ve yatay olarak baęlanmaktadır. Baęlama problemi en kısa yol probleminden yararlanılarak çözülmektedir. En kısa yol problemi kullanılırken yönsüz graf teorisinden yararlanılmaktadır ($G = (V, E)$). Graf teorisindeki düęümler p_i ve p_j olarak ifade edilmektedir. $P_b(p_i)$, p_i pikselinin sınır büyüklüğünü ve $E_{i,j}$ kenar aęırlığını göstermektedir. Bu kenar aęırlığından yararlanılarak dinamik programlama ile baęlantı noktaları birleřtirilmektedir.

$$E_{i,j} = \frac{1}{P_b(p_i) + P_b(p_j)} \quad (3.11)$$

% Kavřak noktalarının başlangıı

gPb haritası kullanılarak sınır haritasının oluřturulması
Bařlangı küme merkezleri belirlenir (Denklem 3.9).

%Kavřak noktalarının yeniden konumlandırılması

for her bir başlangı kavřak noktası

for aday kavřak noktası

f_d = başlangı kavřak noktası ile aday kavřak noktası arasında Öklid mesafesi

Yeni kavřak noktalarının (\hat{p}) belirlenmesi

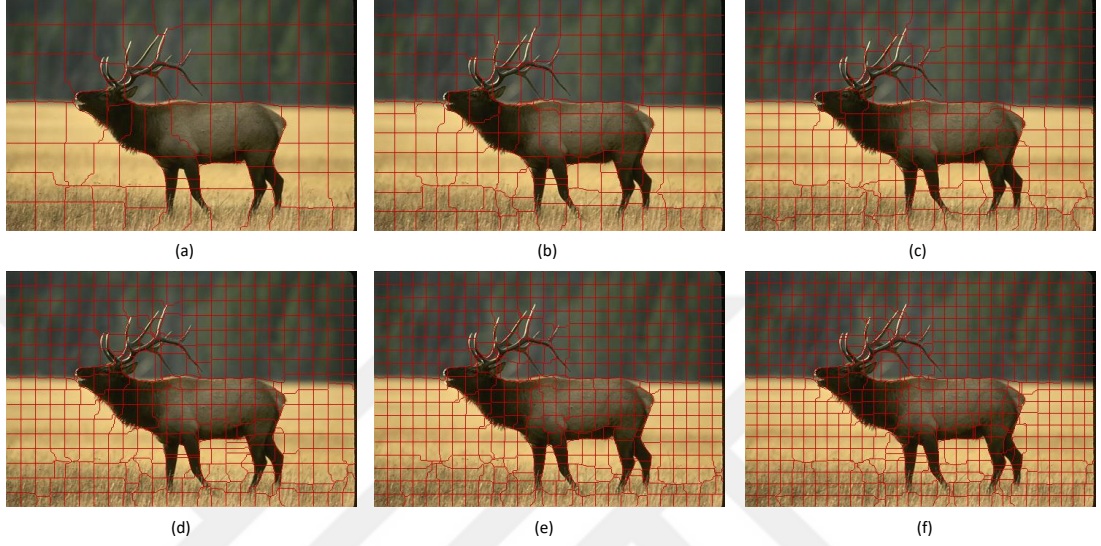
end

end

%Kavřak noktalarının baęlantısı

Kenar aęırlığını hesaplanması ve baęlantının saęlanması

Şekil 3.23'te 100-600 olmak üzere 6 farklı süperpiksel sayısı için RPS süperpiksel algoritmasının performansı verilmiştir. Süperpiksel sayısı arttıkça sınır uyumluluğunun da arttığı görülmektedir. SLIC ve LSC algoritmasına göre daha düzenli süperpikseller oluşturmaktadır.



Şekil 3.23. RPS algoritmasının farklı süperpiksel sayıları için süperpiksel performansları

3.3.4. Entropi Oranı Kümeleme Algoritması (ERC)

Entropi oranı kümeleme (Entropy Rate Clustering, ERC) algoritması graf tabanlı bir algoritmadır ve kümeleme için yeni bir amaç fonksiyonu önerilmiştir. Amaç fonksiyonunun maksimum değerinin bulunması için açgözlü optimizasyonundan yararlanılmıştır. Başlangıç süperpiksel oluşumu için graf teorisinden yararlanılmaktadır. Pikseller $G = (V, E)$ şeklinde tanımlanarak graf uzayına dönüştürülmektedir. Graf uzayında düğüm olarak tanımlanan V değeri imgede piksel olarak tanımlanmaktadır. E değerleri ise pikseller arasında bağlantı kuran kenar yapılarını ifade etmektedir. $A \subseteq E$ olmak üzere K tane ayrık altkümesi açgözlü optimizasyon yöntemiyle $G = (V, A)$ bulunmaktadır. Ayrık zamanlı X rastgele değişkeni, p_x olasılık kütle fonksiyonu olmak üzere $H(X)$ şeklinde entropi tanımlanmaktadır (Denklem 3.12).

$$H(X) = -\sum_{x \in X} p_x(x) \log p_x(x) \quad (3.12)$$

$G = (V, A)$ graf uzayının rastgele yürüme için entropi oranı

$$H(A) = \sum_i \mu_i \sum_j p_{i,j} \log(p_{i,j}(A)) \quad 3.13$$

şeklinde tanımlanmaktadır. Amaç fonksiyonunu oluşturan bir diğer terim ise $B(A)$ denge terimidir. N_A graf uzayında bağlı bileşen sayısı olarak ifade edilmektedir. Z_A bu uzayda küme üyelik dağılımının rastgele değişkeni olarak tanımlanmaktadır.

$$B(A) = H(Z_A) - N_A = -\sum_i p_{Z_A}(i) \log(p_{Z_A}(i)) - N_A \quad (3.14)$$

Entropi oranı ile denge terimi λ denge terimiyle birleşerek amaç fonksiyonu oluşmaktadır.

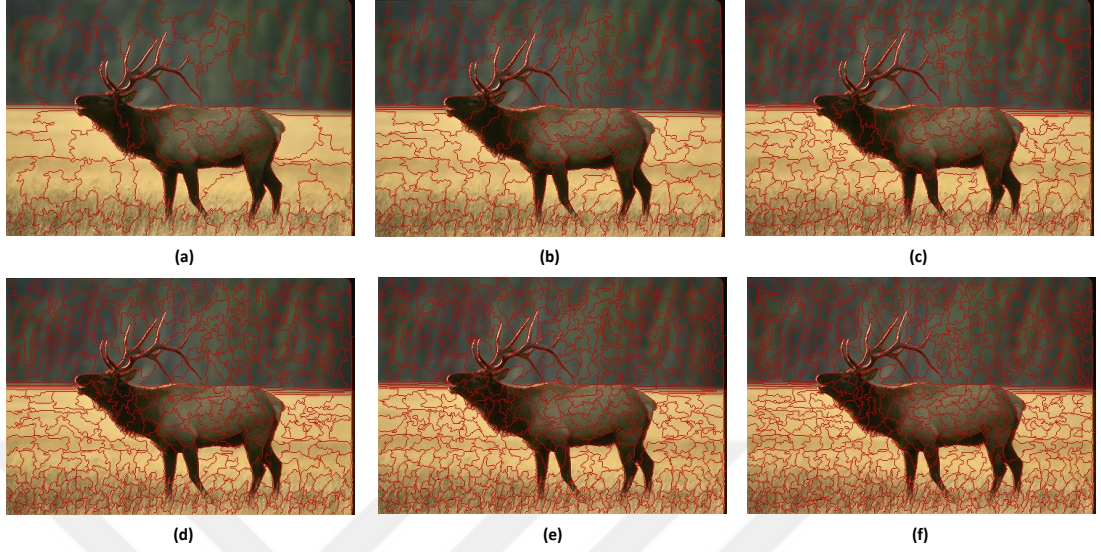
$$\max_A (H(A) + \lambda B(A)) \quad (3.15)$$

Amaç fonksiyonun maksimum değerinin bulunması için aç gözlü optimizasyon algoritmasının sözde kodu aşağıda verilmiştir.

Girdi: $G = (V, E), w: E \rightarrow R^+, K, \lambda$
Çıktı: A
 $A \leftarrow \emptyset, U \leftarrow E$
Repeat
 $\hat{a} \leftarrow \arg \max_{a \in U} F(A \cup \{a\}) - F(A)$
 If $A \cup \{a\} \in I$ **then**
 $A \leftarrow A \cup \{a\}$
 $U \leftarrow U - \{\hat{a}\}$
until

Şekil 3.24'te 100-600 olmak üzere 6 farklı süperpiksel sayısı için ERC süperpiksel algoritmasının performansı verilmiştir. Süperpiksel sayısı arttıkça sınır

uyumluluğunun da arttığı görülmektedir. Bu çalışmada incelenen süperpiksel algoritmalarının arasında en düzensiz süperpiksel oluşturan algoritmadır.



Şekil 3.24. ERC algoritmasının farklı süperpiksel sayıları için süperpiksel performansları

3.3.5. Turbo Piksel Algoritması (TP)

Turbo piksel (Turbopixel, TP) süperpiksel algoritmasında, süperpiksel olarak adlandırılan bir imgenin yoğun bir yansımasını hesaplamak için geometrik akış temelli bir algoritma tanımlanmaktadır. TP algoritmasında süperpiksel bölgesi olarak tanımlanan atanmış bölge ile bu bölgenin dışında kalan atanmamış bölgedeki tüm pikseller ile sınır oluşturma amaçlanmaktadır. Süperpikseller imge üzerinde eşit uzaklıklarda dağılmış daire tohumlarının sınır hız vektörlerine göre açılması ile oluşmaktadır. Her pikselin atanmış ve atanmamış bölge arasındaki en yakın sınır noktasına göre işaretli uzaklığı denklem 3.16'da formülü ile hesaplanmaktadır.

$$\psi_i = -S \|\nabla \psi\| \quad (3.16)$$

Denklemden S sınır noktasının hareket hızını, $\nabla \psi$ uzaklığın zamansal değişimini ifade etmektedir. Bu denkleme göre atanmış bölgedeki piksel uzaklıkları pozitif değerli, atanmamış bölgedeki piksel uzaklıkları ise negatif değerli olmaktadır.

Sınır noktaları ψ uzaklığının sıfır seviye seti ile ifade edilmektedir. Öklid uzaklığının zamanda birinci dereceden ayrıştırılmasıyla denklemi oluşturulmaktadır.

$$\psi^{n+1} = \psi^n - S_I S_B \|\nabla \psi^n\| \Delta t \quad (3.17)$$

Δt sınırların evrimsel olarak oluşması için zaman adımını tanımlamaktadır. S_I yerel imge yapısı ve sınır noktalarının süperpiksel geometrisini, S_B ise sınır noktalarının diğer süperpikseller yakınlığını hesaplamaktadır. Geometrik akış tabanlı algoritma TP için üç özellik gerekmektedir. Bunlardan ilki sınırlara yaklaştıkça sınır büyümeleri yavaşlamalıdır. İkincisi ise sınırlar belirgin olmalı ve oluşan süperpiksellerin sınırları pürüzsüz oluşması gerekmektedir. Üçüncü özellik ise süperpiksellerin çakışmama durumunun gerekmesidir yani her bir pikselle atanan etiket tek olmalıdır.

Girdi: I giriş imgesi, K süperpiksel sayısı

Çıktı: B süperpiksel sınırları

1: I imgesine K tohumları dikdörtgensel olarak yerleştirilir.

2: Tohum pozisyonları yüksek gradyanlı bölgelerden uzaklaştırılır.

3: Tüm tohum pikselleri "atanmış" olarak ayarlanır.

4: "Atanmış" bölgelerde ψ^0 öklid uzaklığı hesaplanır.

5: atanan_pikseller $\leftarrow \sum_{x,y} (\psi^0(x,y) \geq 0)$ yazılır.

6: $\phi(x,y)$ piksel yakınlığı hesaplanır.

7: $n \leftarrow 0$

8: **while** atanan_pikseller değişim büyük olduğunda

9: İmge hızı S_I hesaplanır.

10: Sınır hızı S_B hesaplanır.

11: $S \leftarrow S_I S_B$

12: ψ^n sıfır seviye setinin yakınında dar bir bantta S hız genişletilir.

13: ψ^{n+1} hesaplanır.

14: $n \leftarrow n + 1$

15: atanan_pikseller $\leftarrow \sum_{x,y} (\psi^0(x,y) \geq 0)$ yazılır.

16: $B \leftarrow \psi^n$ uzamdaş iskelet atanır.

17: **return** B

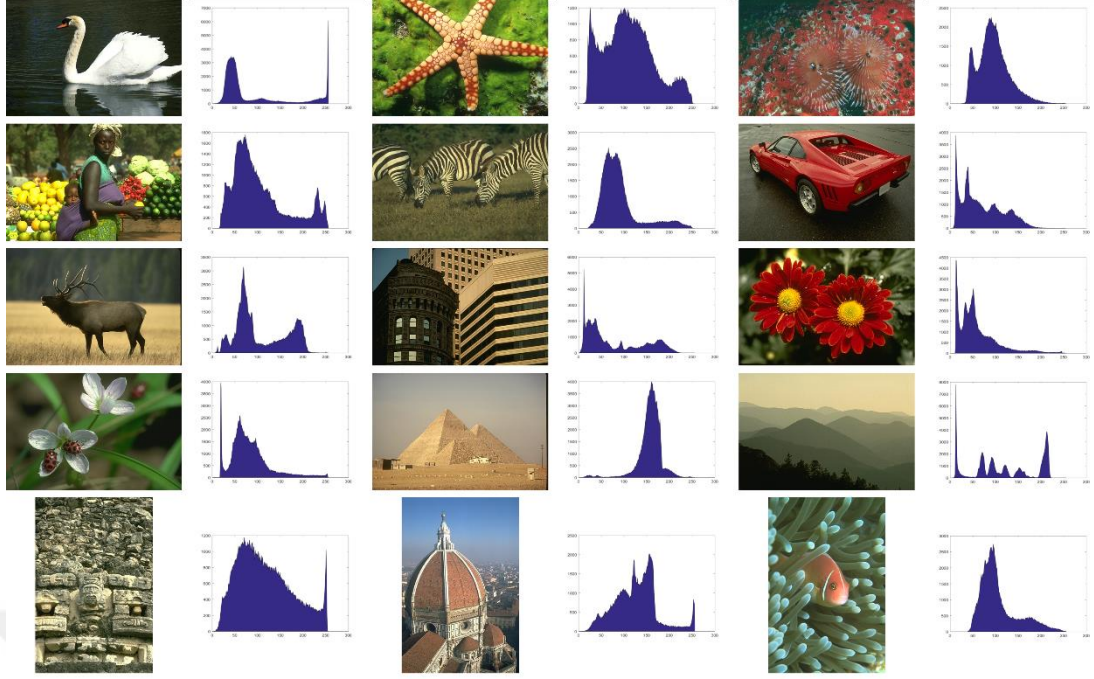
TP için sözde kod yukarıda verilmiştir. Algoritma sonucunda tekdüze ve derlitoplu süperpikseller oluşmaktadır (Şekil 3. 25).



Şekil 3.25. TP algoritmasının farklı süperpiksel sayıları için süperpiksel performansları

3.4. Test İmgeleri

Bu tez çalışmasında süperpiksel algoritmalarının performanslarını ölçebilmek amacıyla literatürdeki diğer bölütleme ve süperpiksel çalışmalarında da en yaygın şekilde kullanılan Berkeley Bölütleme Veri Kümesi kullanılmıştır. (<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>). Bu veri kümesi BSD300 denilen 200 eğitim ve 100 test imgelerinden oluşmaktadır. Ayrıca bu BSD300 verisinin yanı sıra 200 doğrulama imgesi ile BSD500 veri kümesi bulunmaktadır. Bu çalışma boyunca BSD500 veri kümesinden farklı kontrast, renk ve doku davranışlarına sahip Şekil 3.26'da verilen 15 adet imge kullanılmıştır. Şekil 3.26'da verilen imgelerin histogramları oluşturulurken imgeler gri seviyeli imgeye dönüştürülerek elde edilmiştir. Bu imgelerin boyutları 321×481 veya 481×321 şeklindedir.



Şekil 3.26. Tez çalışmasında kullanılan test imgeleri

4. BAŞARI METRİKLERİ

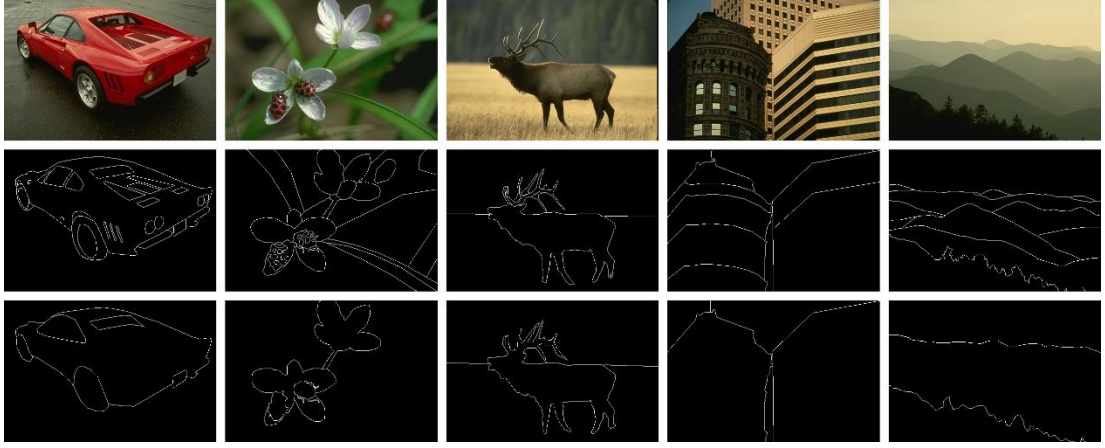
Süperpiksel algoritmalarının başarılarını ölçebilmek için literatürde çeşitli performans metrikleri önerilmiştir. Bu metrikler oluşturulan süperpiksellerin şekli, homojenliği ve sınır uyuşma durumlarını ele alarak incelemektedir. Süperpiksellerin doğruluğunu ölçebilmek için sınır örtüşmesi ($SÖ$), eksik bölütleme hatası (EBH), erişilebilir bölütleme doğruluğu (EBD), açıklık değişimi (AD), doluluk derecesi (DD), boyut değişimi (BD) ve küme içi değişimi ($KİD$) metrikleri kullanılmaktadır. Ayrıca imgelerin kalitesini değerlendirebilmek için de $PSNR$ metriklerinden yararlanılmıştır.

4.1. Kesin Referans

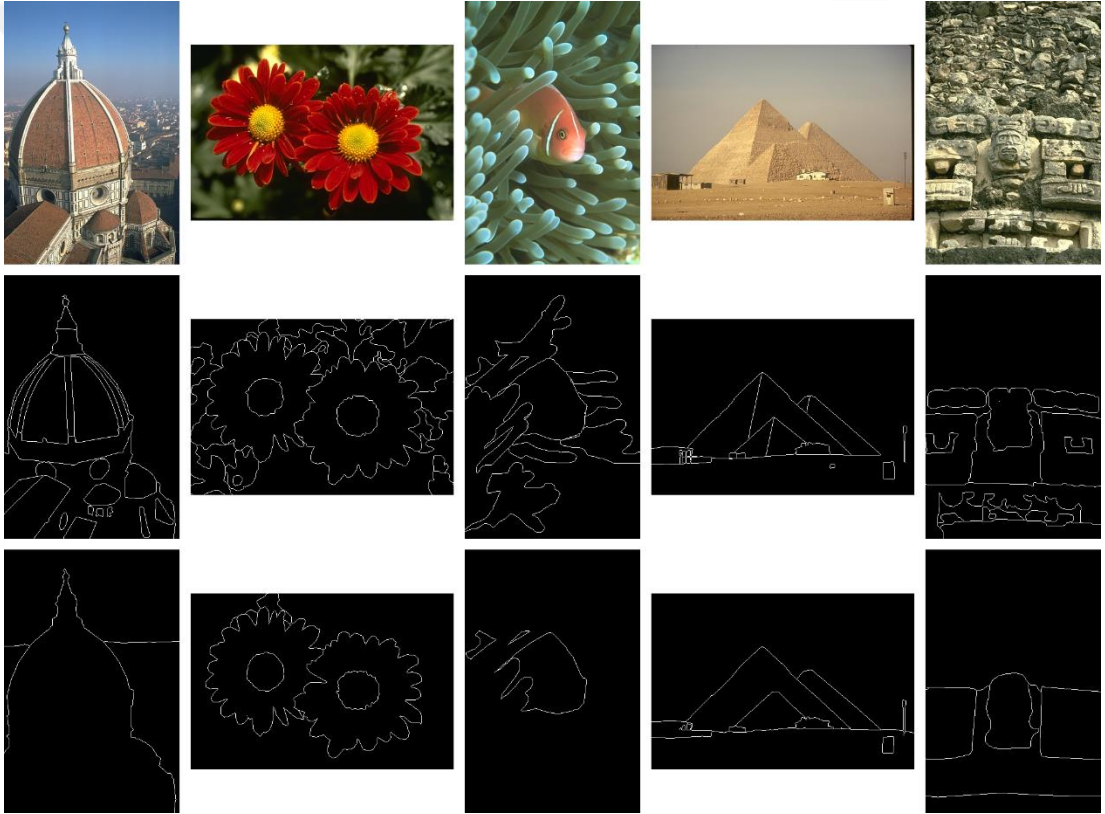
Süperpiksel algoritmalarının performanslarını ölçmek amacıyla, kullanılan imgelerin gerçek sınır bilgilerini içeren kesin referans verisi kullanılmıştır. Bu veri kullanılan veri kümesinde her bir imge için kullanıcılara sunulmuştur. Her bir imge için farklı segment sayılarını içeren kesin referans verisi bulunmaktadır. Şekil 4.1-3'te verilen test imgeler ve onlara ait minimum ve maksimum segmente sahip kesin referans imgeleri verilmiştir. Süperpiksel algoritmalarınca elde edilen süperpiksellerin sahip olduğu içerik ve sınır bilgilerinin kesin referans verisindeki içerik ve sınır bilgisiyle olan yakınlık ve benzerlik ilişkileri analiz edilerek süperpiksellerin başarıları değerlendirilmektedir.



Şekil 4.1. Test imgeleri ve kesin referans gösterimi-1



Şekil 4.2. Test imgeleri ve kesin referans gösterimi-2



Şekil 4.3. Test imgeleri ve kesin referans gösterimi-3

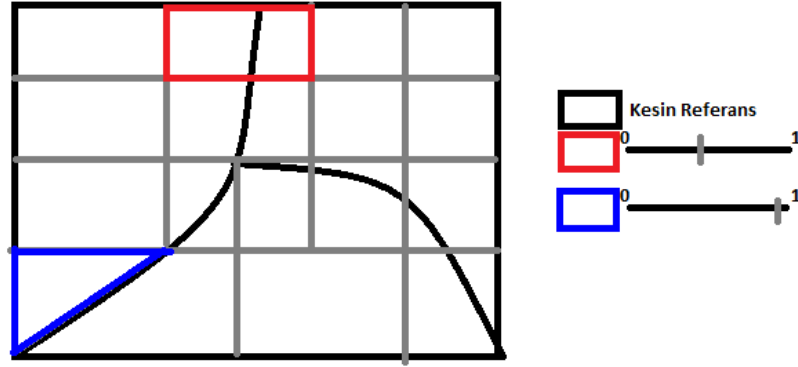
4.2. Sınır Örtüşmesi

Sınır örtüşmesi ($SÖ$), süperpiksel sınırları ile kesin referansın oluşturduğu sınırların birbirine yakınlık derecesinin ifadesidir. En ideal durum her iki bölütleme sonucunda sınırlarının örtüşmesidir. p kesin referansın oluşturduğu bölütlerin sınır piksellerinin, q ise süperpiksellerin oluşturduğu sınır

pikselleridir. $dist_{xy}$, sınırlar arasında öklid mesafesi hesaplanmaktadır ve ε değeri 2 olarak seçilmiştir. Sınır örtüşmesi değeri $[0,1]$ aralığında olup süperpiksel sayısı arttıkça 1'e yaklaşmaktadır. İdeal durumda 1 çıkması gerekmektedir.

$$S\ddot{O}_G(S) = \frac{\sum_{p \in \delta G} \phi(\min_{q \in \delta S} dist_{xy}(p,q) < \varepsilon)}{\delta G} \quad (4.1)$$

Şekil 4.4'te başarı metriklerinden sınır örtüşmesinin gösterimi verilmiştir. Bu şekilde siyah çizgi kesin referansı ve gri kutular süperpikselleri göstermektedir. Burada kırmızı ve mavi süperpikselin sınır örtüşmesi karşılaştırıldığında mavi süperpikselin sınır örtüşmesi değeri 1'e yakındır. Çünkü kenar uyumluluğu daha yüksektir.



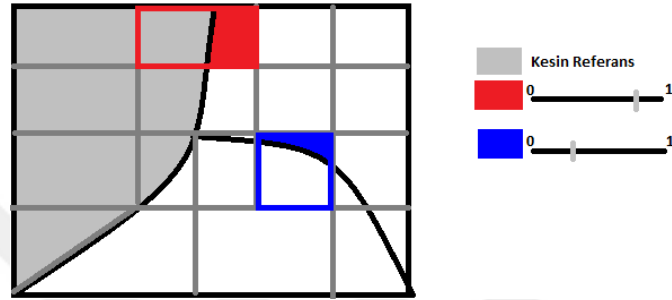
Şekil 4.4. Sınır örtüşmesi için sentetik gösterim

4.3. Eksik Bölütleme Hatası

Eksik bölütleme hatası (EBH), kesin referans bölgeleri ile süperpiksel bölgeleri arasında sızıntı bölgeleri ölçer yani örtüşmeyen bölgeleri hesaplar. Süperpiksel $S = \{S_1, S_2, \dots, S_j\}$, kesin referans $G = \{G_1, G_2, \dots, G_i\}$ ile gösterilmektedir. N imgedeki toplam piksel sayısıdır. Eksik bölütleme hatası $[0,1]$ aralığında olup, ideal durumda 0 olması gerekmektedir. Süperpiksel sayısı arttıkça eksik bölütleme hata değeri düşmektedir.

$$EBH(S, G) = \frac{1}{N} \sum_{G_i \in G} \sum_{S_j \cap G_i \neq \emptyset} \min\{|S_j \cap G_i|, |S_j - G_i|\} \quad (4.2)$$

Şekil 4.5'te eksik bölütleme hatasının sentetik gösterimi verilmiştir. Burada gri bölge kesin referans bölgesini, kırmızı ve mavi bölgelerde süperpiksellerin sızıntı bölgelerini ifade etmektedir. EBH , süperpiksellerin sızıntı bölgelerini ölçmektedir ve ne kadar az sızıntı bölge olursa süperpiksel algoritmasının başarısı yüksek olmaktadır. Mavi ve kırmızı süperpiksellerin sızıntı bölgeleri karşılaştırıldığında mavi süperpikselin sızıntı bölgesinin daha az olduğundan EBH değeri daha düşüktür.



Şekil 4.5. Eksik bölütleme hatası için sentetik gösterim

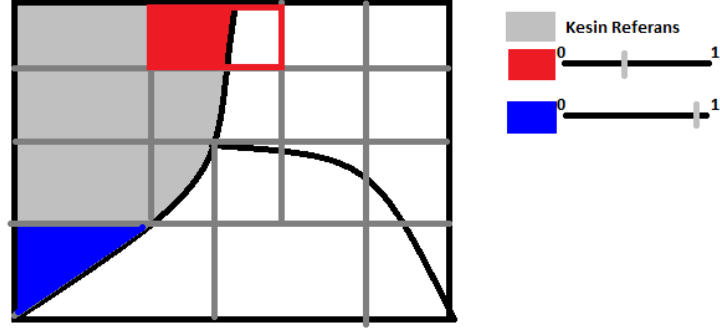
4.4. Erişilebilir Bölütleme Doğruluğu

Erişilebilir bölütleme doğruluğu (EBD), her bir süperpikselin etiketi ile kesin referansın etiketinin örtüşme alanını hesaplar. Bu parametre imgedeki nesneyi doğru tanıyıp tanımadığının ölçüsüdür. Süperpiksel $S = \{S_1, S_2, \dots, S_j\}$, kesin referans $G = \{G_1, G_2, \dots, G_i\}$ ile gösterilmektedir. N toplam piksel sayısıdır. Erişilebilir bölütleme doğruluğunun değeri $[0,1]$ arasında değişmektedir. Süperpiksel sayısı arttıkça erişilebilir bölütleme doğruluğu artmaktadır yani 1'e yaklaşmaktadır.

$$EBD(S, G) = \frac{1}{N} \sum_{S_j \in S} \max_{G_i} \{|S_j \cap G_i|\} \quad (4.3)$$

$$S_j \cap G_i = \{x \in S_j \mid x \in G_i\}$$

Şekil 4.6'da başarı metriklerinden erişilebilir bölütleme doğruluğu verilmiştir. EBD , kesin referans ile süperpikselin örtüşme alanı hesapladığından mavi ve kırmızı süperpiksel alanları karşılaştırılmıştır. Mavi süperpikselin tamamen kesin referansla örtüştüğünden EBD değerinin maksimum değeri olan 1'e daha yakındır.



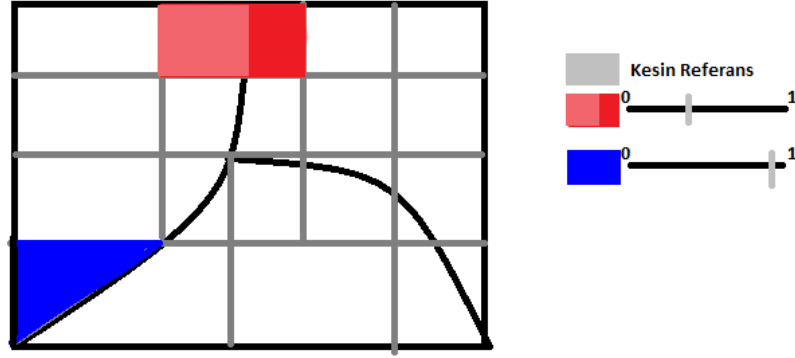
Şekil 4.6. Erişilebilir bölütleme doğruluğu için sentetik gösterim

4.5. Açıklık Değişimi

Açıklık değişimi (AD), süperpiksellerden kaynaklanan renk bozukluk seviyesini ölçmektedir yani süperpikseller ile orijinal pikseller arasındaki tüm farklılıkları değerlendirmektedir. μ_k her bir süperpikselin piksel yoğunluğunun değerlerinin ortalamasını ifade etmektedir. μ değeri ise tüm imgenin ortalama değeridir. x_i imgedeki her bir pikselin yoğunluğunu ifade etmektedir. Açıklık değişimi $[0,1]$ aralığında olup ideal durumda 1 olması gerekmektedir ve süperpiksel sayısı arttıkça açıklık değişimi değeri artmaktadır.

$$AD = \frac{\sum_{k=1}^N (\mu_k - \mu)^2}{\sum_{i \in I} (x_i - \mu)^2} \quad (4.4)$$

Şekil 4.7'de açıklık değişimi gösterimi verilmiştir. Açıklık değişimi oluşan süperpiksellerin homojenliğini ile hesaplanmaktadır. Burada mavi ve kırmızı süperpikseller incelendiğinde kırmızı süperpikselin homojen olmadığı gözlemlenmektedir ve açıklık değişimi değeri daha düşüktür. Mavi süperpiksel homojen piksellerden oluştuğu için açıklık değişimi değeri daha yüksektir.



Şekil 4.7. Açıklık değişimi için sentetik gösterim

4.6. Doluluk Derecesi (Derlitopluluk)

Bir şeklin kompaktlığını ölçmek için matematikte yaygın olarak kullanılan izoperimetrik katsayısının değerinin hesaplanmasıyla bulunmaktadır. İzoperimetrik katsayı problemi iki boyutlu bir şeklin çevrelediği sınır uzunluğunun en büyük alana sahip olması şeklinde ifade edilmektedir. Bu problemin çözülebilmesi için de çevrelediği sınır ile alanının eşit olabilmesi için şeklin daire olması gerekmektedir. Bu durumda denklem 4.5'te S süperpikselin alanını, L ise süperpikselin çevresini ifade etmektedir. N ise süperpiksel sayısını ifade etmektedir. Doluluk derecesi 1'e ne kadar yakın olursa oluşturulan süperpiksellerin o kadar düzenli şekil ve boyutta olduğunu ifade etmektedir.

$$DD = \frac{1}{N} \sum_{i=1}^N \frac{4\pi S(s_i)}{L^2(s_i)} \quad (4.5)$$

4.7. Boyut Değişimi

Boyut değişim (BD), süperpiksellerin boyut dağılımlarını ölçmek ve topolojilerini korumak amacıyla kullanılan bir metriktir. Boyut değişimi hesaplanırken süperpiksellerin sınır boyutunun varyans değeri hesaplanmaktadır. Boyut değişimi değeri ne kadar düşük olursa oluşturulan süperpiksellerin birbirine boyut açısından o kadar benzediği ifade edilmektedir.

4.8. Küme İçi Değişimi

Küme içi değişimi (KID), her süperpiksel içindeki standart sapmayı hesaplayarak bölütlemenin kalitesini ölçmektedir. İyi bölütleme, her süperpiksel içerisinde küçük farklılıklar gösteren homojen kümeler oluşturmalıdır. Küme içi değişimi hesaplamak için renk uzayının her kanalı için standart sapma hesaplanarak ortalama değer hesaplanmaktadır. $|S|$ imgedeki süperpiksel sayısını ve $|s|$ süperpiksel içerisindeki piksel sayısını ifade etmektedir. μ_s süperpikselin ortalama değerini ve p süperpiksel içerisindeki pikselleri ifade etmektedir.

$$KID = \frac{1}{|S|} \sum_{s \in S} \sqrt{\frac{\sum_{p \in s} (p - \mu_s)^2}{|s|}} \quad (4.6)$$

4.9. PSNR

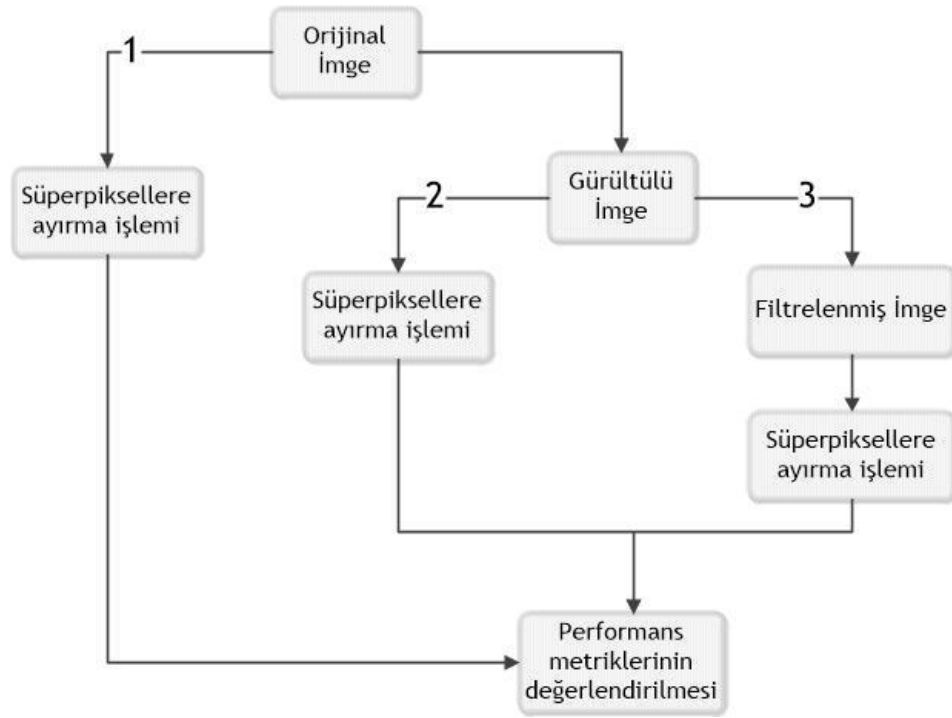
Tepe sinyal gürültü oranı (Peak Signal to Noise Ratio, $PSNR$) imge kalitesini ölçmek için kullanılan bir metriktir. Orijinal imge ile gürültülü imge arasındaki benzerlik kalitesini hesaplamaktadır. Hesaplama sonucunda $PSNR$ tek bir değer üretmektedir.

$$PSNR = 10 \log \frac{I_{\max}^2}{MSE} (dB) \quad (4.7)$$
$$MSE = \frac{1}{MN} \sum_{i=0}^M \sum_{j=0}^N (Y_{i,j} - S_{i,j})^2$$

I_{\max} imgedeki maksimum gri seviye değeri ve MSE ortalama karesel hata değerini ifade etmektedir. M ve N imgenin sırasıyla satır ve sütun sayısıdır. $S_{i,j}$ orijinal imgenin piksel değerini ve $Y_{i,j}$ içeriği değiştirilmiş imgenin piksel değerlerini göstermektedir. Bu değerlerin yüksek olması kalitesinin de yüksek olduğu anlamına gelmektedir. Yani iki resim aynı olduğunda $PSNR$ değeri sonsuzdur.

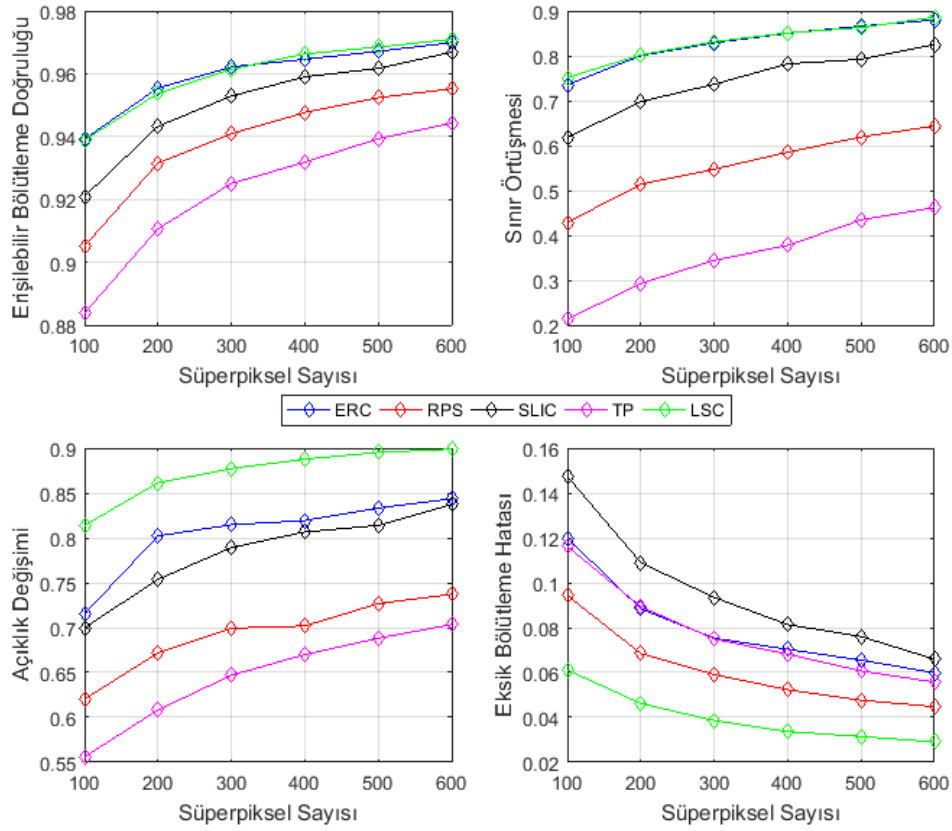
5. SONUÇLAR

Süperpikseller bilgisayarlı görü sistemlerinde, medikal imge bölütlemelerinde sıklıkla kullanılmaktadır. Bu uygulamalarda kullanıldığında süperpiksel algoritmalarının performansı çeşitli analizler ile değerlendirilmektedir. Tez çalışması Şekil 5.1'de verilen blok diyagramda gösterildiği gibi 3 ana aşamadan oluşmaktadır. İlk aşama orijinal test imgelerine literatürde en yaygın olarak kullanılan 5 süperpiksel algoritması uygulanmaktadır. Tezin ikinci aşaması ise iletim ortamından veya biyomedikal cihazlardan kaynaklı gürültülerin etkisini süperpiksel algoritmaları üzerinde incelemek için orijinal test imgelerine farklı gürültü tipleri ve seviyelerinde gürültü eklenerek süperpiksel algoritmalarının performansları değerlendirilmiştir. Tez çalışmasının son aşaması ise gürültülü imgelere filtre uygulanarak orijinal imgedeki süperpiksel performansı ile karşılaştırılmıştır. Çalışmalar sırasında 15 test imgesi kullanılmıştır ve test imgelerine ait minimum 5 ve maksimum 7 keskin referans bölütleri üzerinde işlem yapılmıştır. Aşağıda verilen sonuçlar hem kesin referans değerlerinin ortalama değeri hem de 15 test imgesinin ortalama değeridir.



Şekil 5.1. Tez çalışmasının genel blok diyagramı

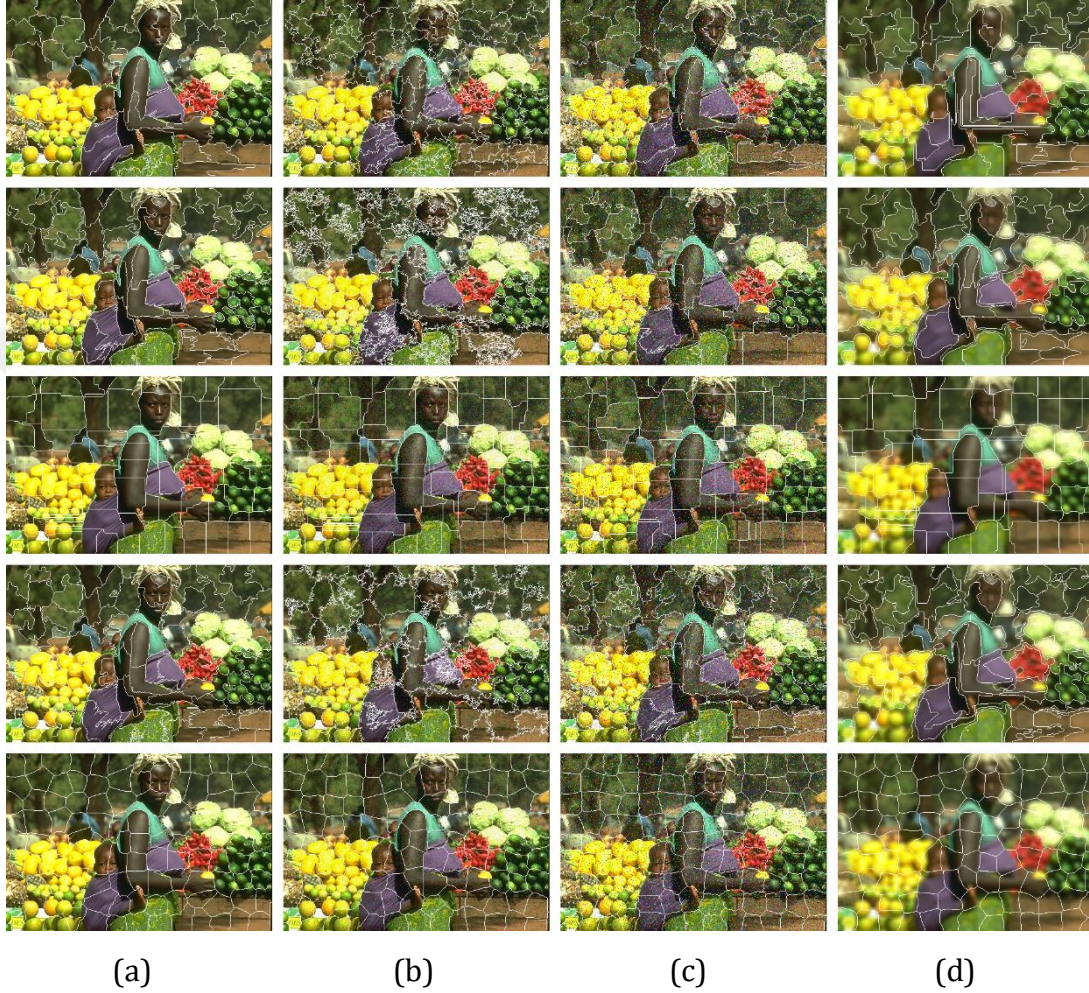
Tez çalışmasının ilk aşaması orijinal imgeler üzerinde literatürde yaygın olarak kullanılan 5 süperpiksel algoritmasının başarı metrikleri ile karşılaştırılmasıdır. Şekil 5.2'de orijinal imgelerin süperpiksel sayısına bağlı olarak başarı metrik grafikleri verilmiştir. Grafikler incelendiğinde orijinal imgelerde en iyi performansı LSC algoritması sunmaktadır. Burada en kötü performansı EBD, SÖ ve AD başarı metriklerinde TP algoritması olmasına rağmen EBH başarı metriğinde ise SLIC algoritmasındadır. Süperpiksel sayısının artmasıyla başarı metrikleri de artmaktadır.



Şekil 5.2. Orijinal imgeler için süperpiksel sayısına bağlı olarak algoritmaların performanslarının değerlendirilmesi

Tezin ikinci aşaması gürültülü imgeler için süperpiksel algoritmalarının performansının değerlendirilmesidir. Burada gauss gürültüsü, tuz&biber gürültüsü ve bulanıklaştırma orijinal imgelere farklı seviyelerde uygulanmıştır. Şekil 5.3'te 5 algoritmanın 100 süperpiksel sayısı için gösterimi verilmiştir. İlk satır ERC, ikinci satır LSC, üçüncü satır RPS, dördüncü satır SLIC ve son satır TP süperpiksel algoritmalarını ifade etmektedir. Şekil 5.3(a) orijinal imgeler

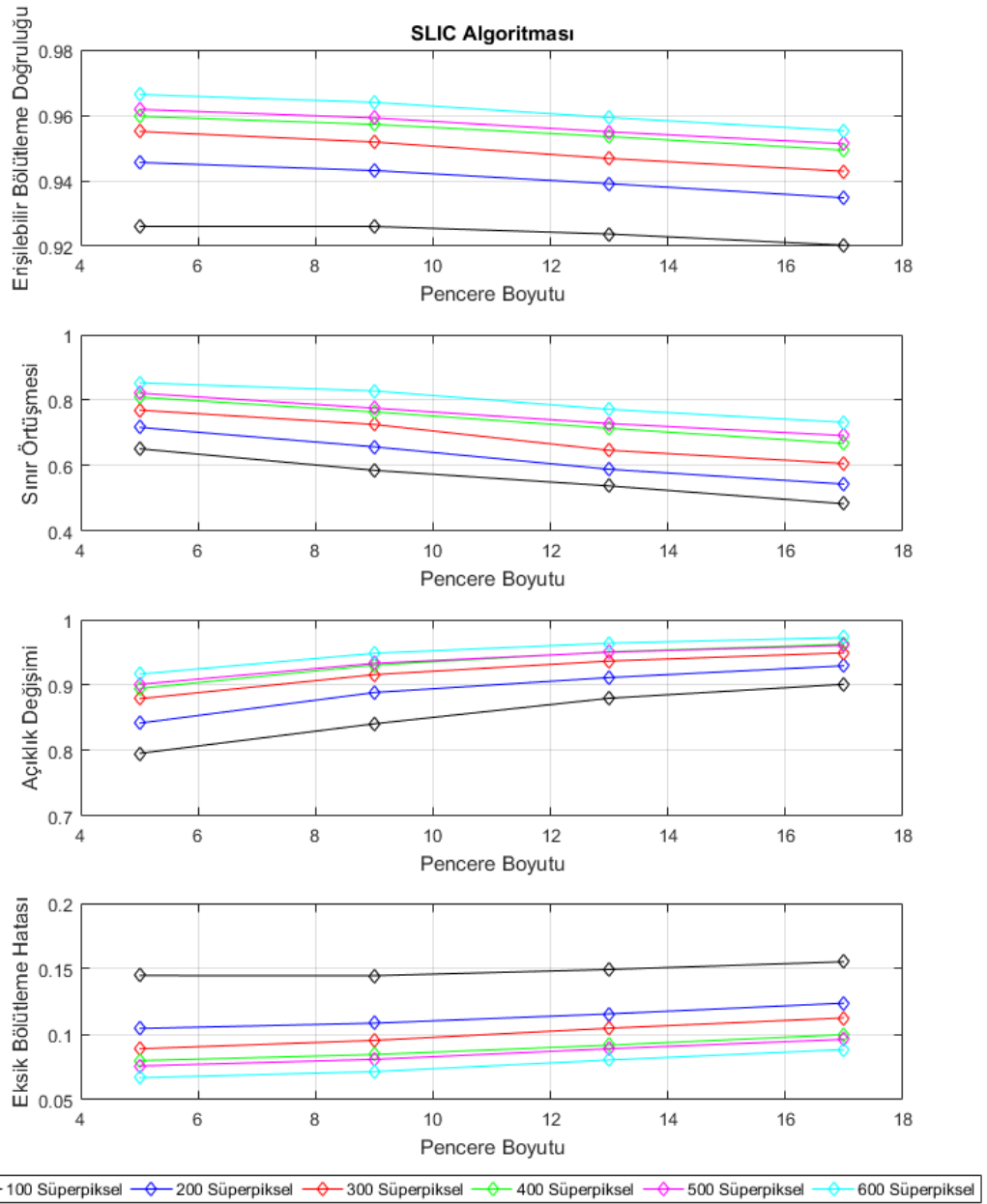
üzerinde süperpiksel performansını, Şekil 5.3(b) gauss gürültülü imgeler üzerinde süperpiksel performansını, Şekil 5.3(c) tuz&biber gürültülü imge üzerinde süperpiksel performansını ve Şekil5.3(d) bulanıklaştırılmış imgeler üzerinde süperpiksel performansı verilmiştir.



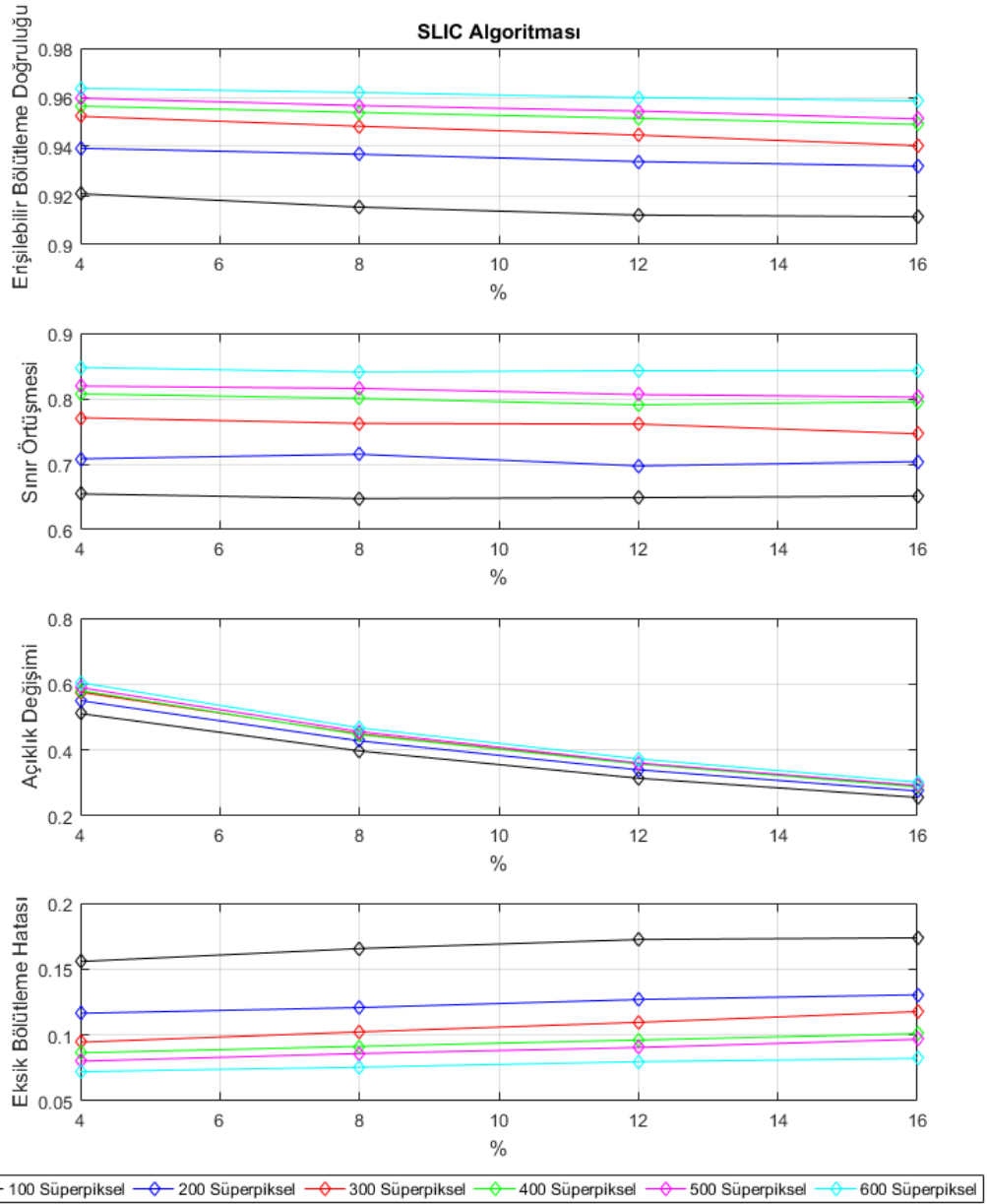
Şekil 5.3. Süperpiksel algoritmalarının 100 süperpiksel için gürültüye bağlı olarak süperpiksel performanslarının incelenmesi (a) Orijinal imge, (b) Gauss gürültülü imge, (c) Tuz&biber gürültülü imge, (d) Bulanıklaştırılmış imge

Şekil 5.4-5.6'da verilen grafikler SLIC algoritmasının gürültü bağışıklığının incelenmesi için başarı metriklerinin sonuçlarını içermektedir. Şekil 5.4'te SLIC algoritmasının bulanıklaştırma etkisi incelenmiştir. Bulanıklaştırma etkisi arttıkça imge piksel değerleri birbirine yaklaşmaktadır. Bu yüzden AD değeri süperpiksellerin homojenliğini incelediği için artmaktadır. SÖ değeri sınır değerlerindeki belirginlik azaldığı için azalmaktadır. EBD değeri de sınırlar

içerisinde kalan piksellerin kesin referans değerine göre farklılaştığı için değeri azalmaktadır. EBH değeri hata parametresi olduğundan bölütleme performansı düştüğünde artmaktadır. Şekil 5.5 ve Şekil 5.6'da sırasıyla gauss ve tuz&biber gürültüsünün SLIC algoritmasına etkisi incelenmiştir. Burada eklenen gürültüler orijinal imgenin içeriğini bozduğu gözlemlenmiştir ve başarı metrikleri oldukça düşmektedir. Üç gürültü tipinde de süperpiksel sayısı arttıkça performansı artarken gürültü seviyesi arttıkça performansının azaldığı sonucuna varılmıştır.



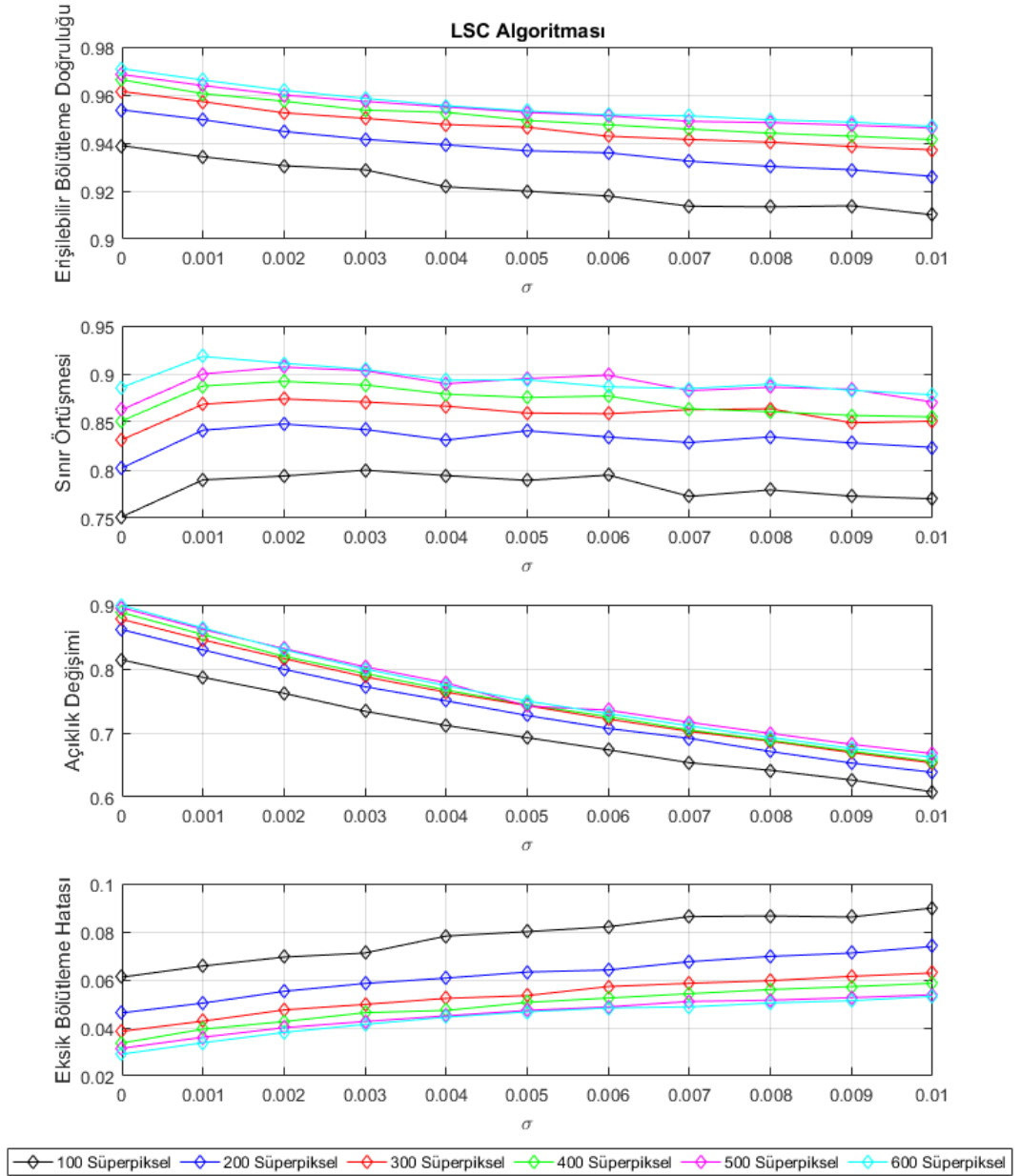
Şekil 5.4. SLIC algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi



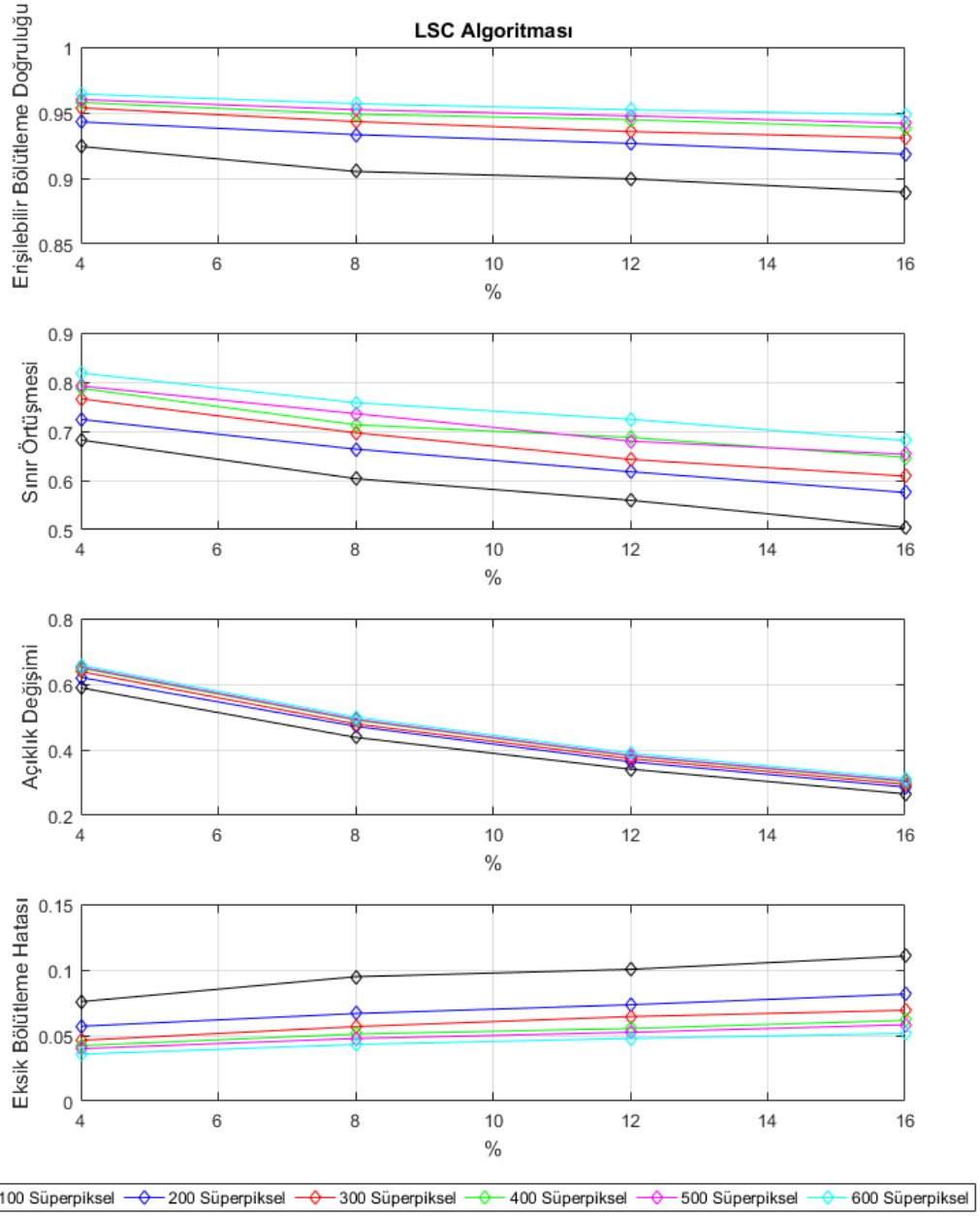
Şekil 5.6. SLIC algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi

Şekil 5.7-5.9'da verilen grafikler LSC algoritmasının gürültü bağımsızlığının incelenmesi için başarı metriklerinin sonuçlarını içermektedir. Şekil 5.7'de gauss gürültüsü için farklı seviyelerdeki başarı metrikleri verilmiştir. Burada süperpiksel sayısı arttıkça algoritmanın performansı da artmaktadır. Ayrıca gürültü seviyesi arttıkça tüm başarı metriklerinde performansı düşmektedir. Şekil 5.8'de tuz&biber gürültüsünün LSC algoritmasına etkisi incelenmiştir. Gürültü seviyesi arttıkça başarı metriklerinin değerleri imgenin yapısının bozulmasından dolayı düşmektedir. Süperpiksel sayısı arttıkça başarı metriklerinin değerleri de artmaktadır. Şekil 5.9'da imgelerin bulanıklaştırılması ile elde edilen LSC algoritmasının başarı metriklerinin sonuçları verilmiştir. Burada gauss ve tuz&biber gürültüsünden farklı olarak bulanıklaştırma etkisi

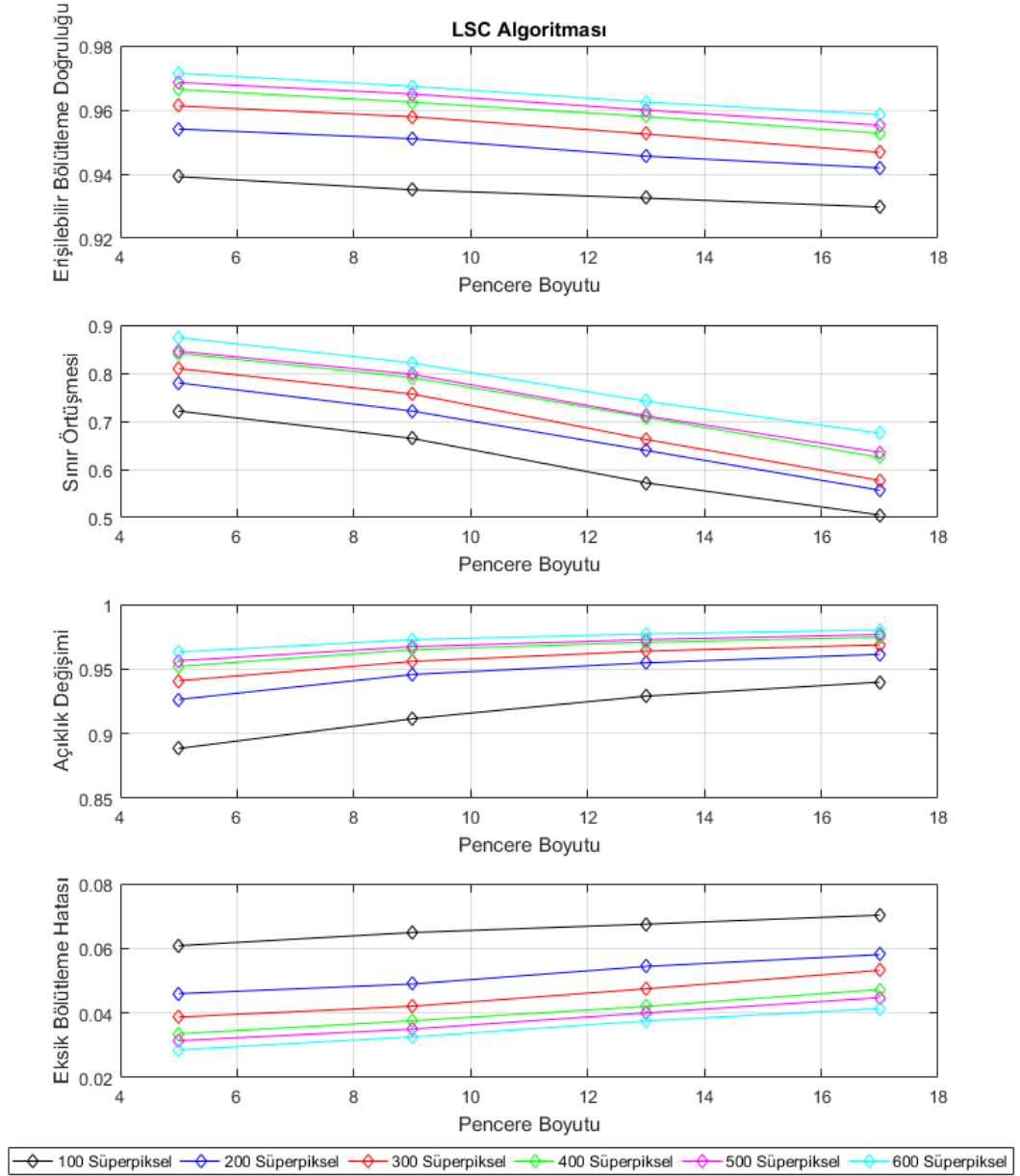
artıkça açıklık değişiminin arttığı gözlemlenmektedir. Bunun nedeni ise bulanıklaştırma ile imgedeki piksel değerlerinin birbirine yoğunluk değeri bakımından yaklaşmasıdır. Böylece her bir süperpiksel içerisindeki piksel değerleri birbirine yaklaşmaktadır ve varyans değeri düşerek açıklık değişimi değerini arttırmaktadır.



Şekil 5.7. LSC algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



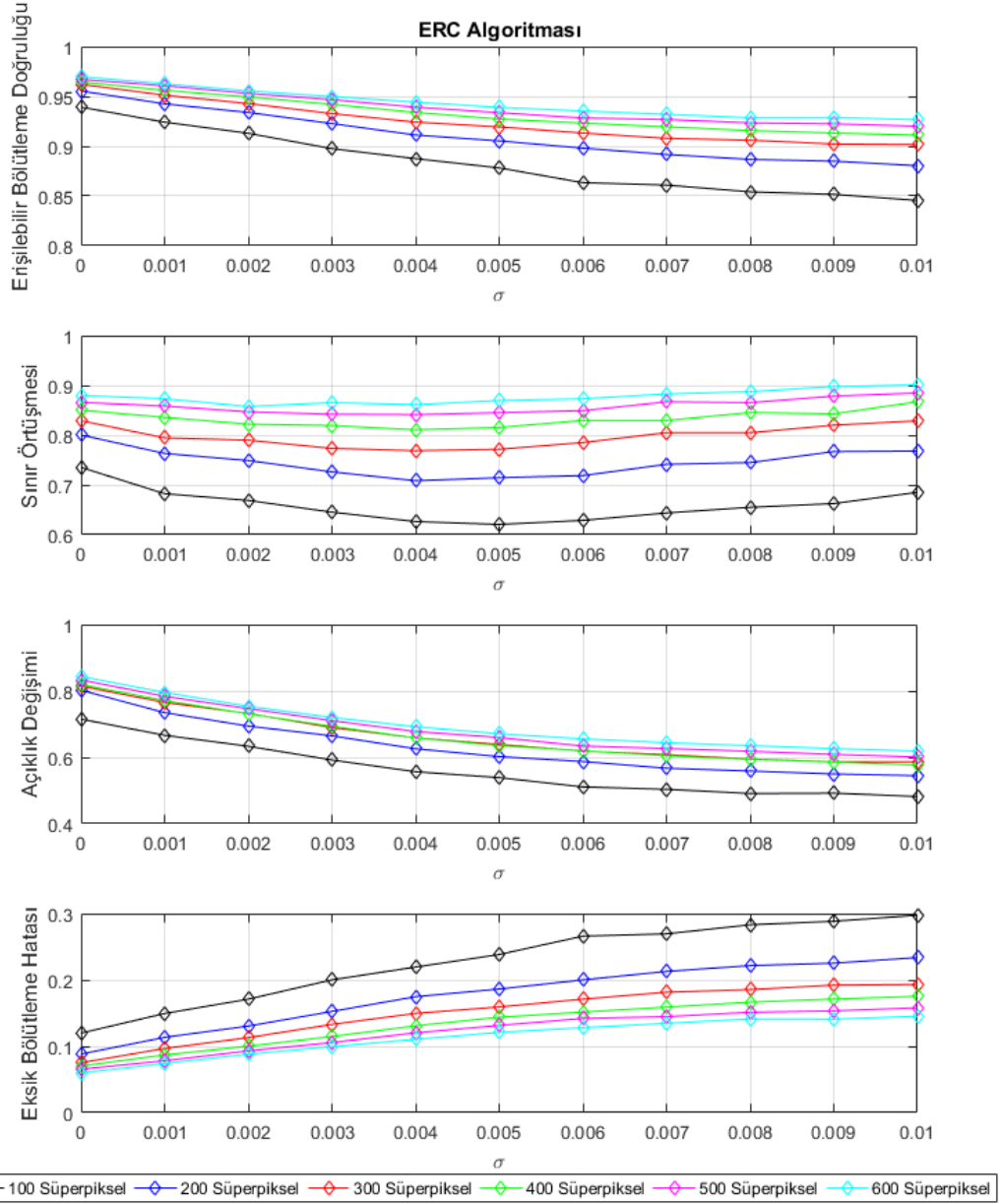
Şekil 5.8. LSC algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



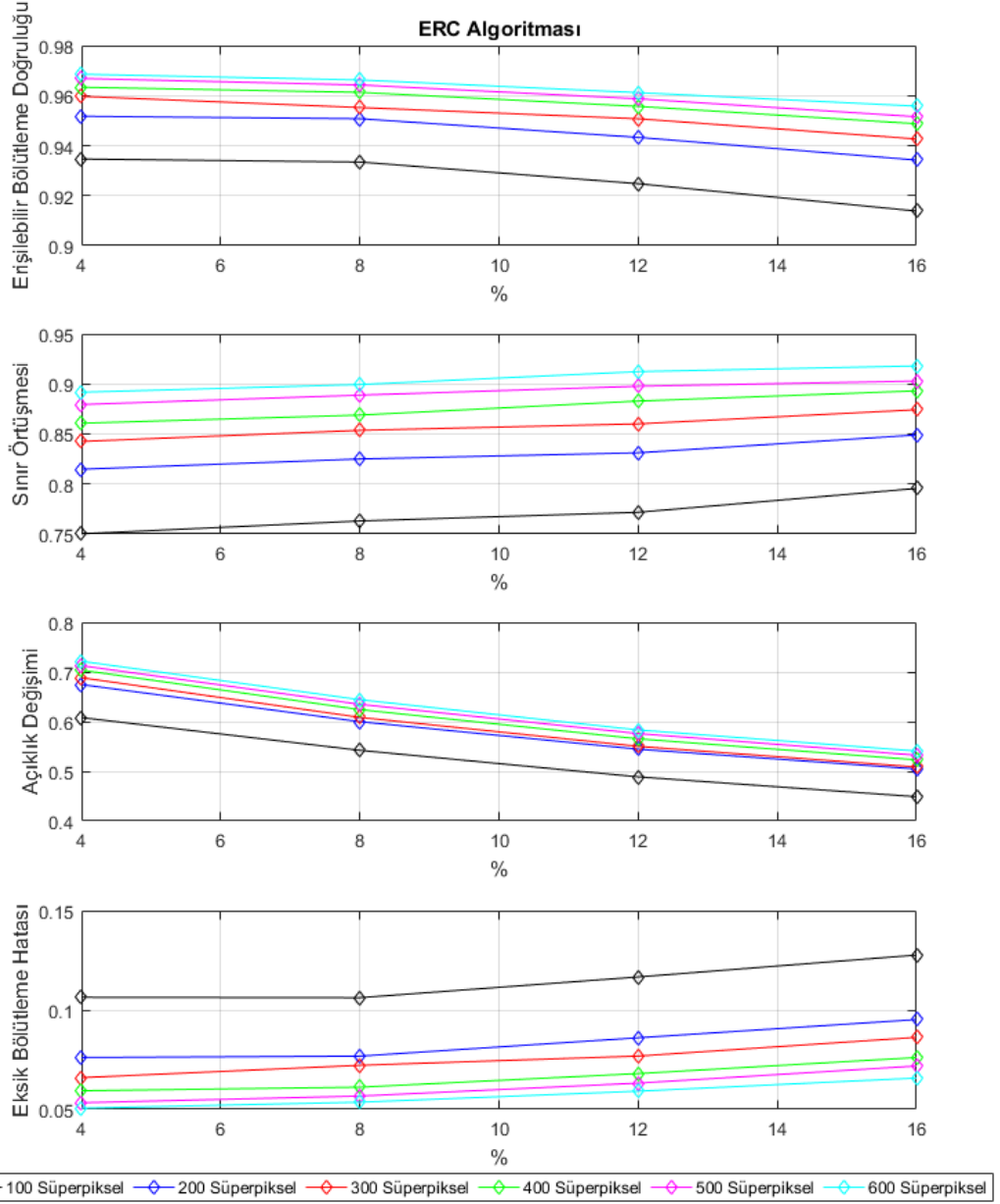
Şekil 5.9. LSC algoritmasının bulanıklaştırma etkisi bağlı olarak metrik değerlerinin incelenmesi

Şekil 5.10-5.12’de verilen grafikler ERC algoritmasının gürültü bağışıklığının incelenmesi için başarı metriklerinin sonuçlarını içermektedir. Şekil 5.10 ve Şekil 5.11’de sırasıyla gauss ve tuz&biber gürültüsünün ERC algoritmasına etkisi incelenmiştir. Burada eklenen gürültüler orijinal imgenin içeriğini bozduğu gözlemlenmiştir ve başarı metrikleri oldukça düşmektedir. Şekil 5.12’de ERC algoritmasının bulanıklaştırma etkisi incelenmiştir. Bulanıklaştırma etkisi arttıkça imge piksel değerleri birbirine yaklaşmaktadır. Bu yüzden AD değeri süperpikselerin homojenliğini incelediği için artmaktadır. SÖ değeri sınır

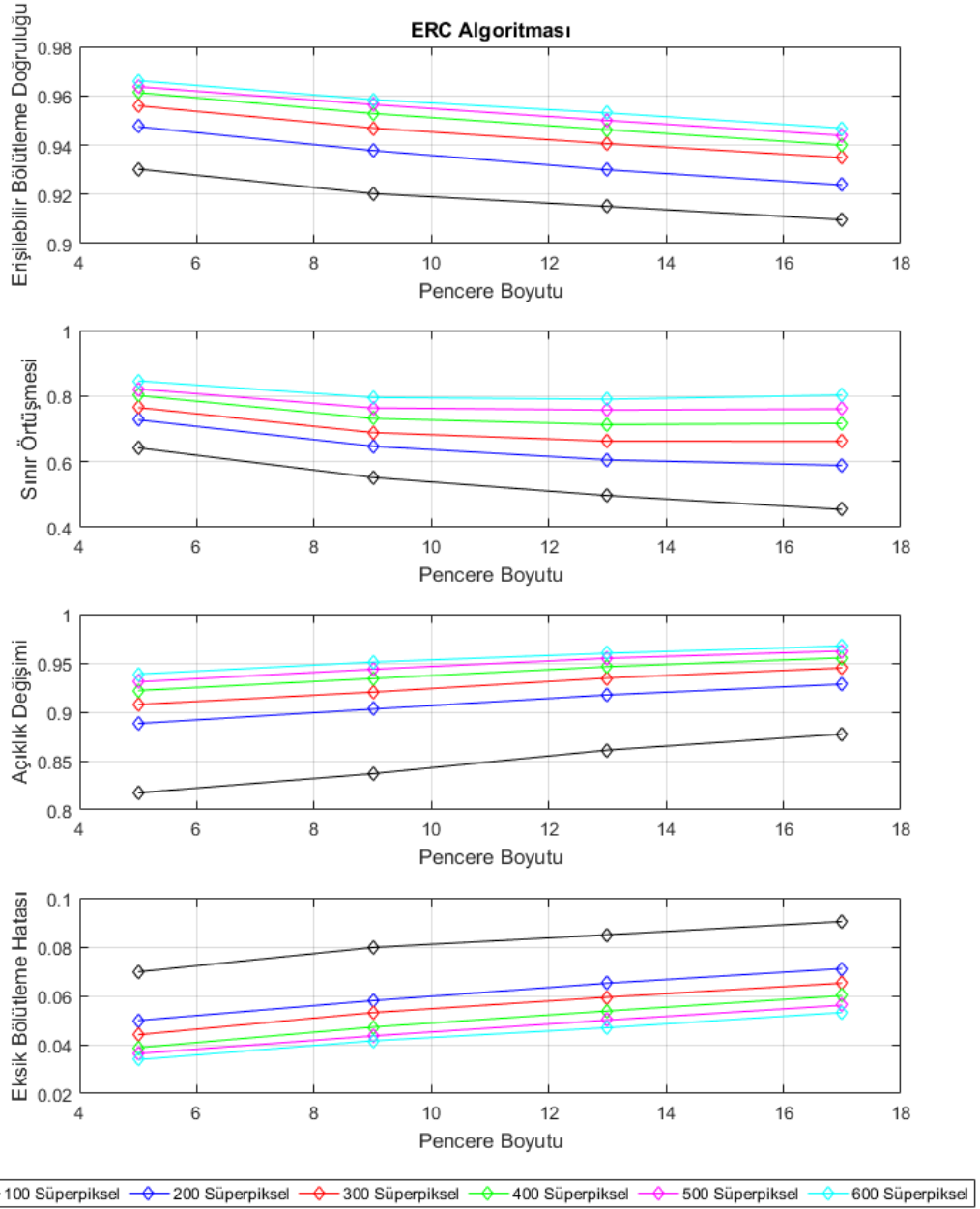
değerlerindeki belirginlik azaldığı için azalmaktadır. EBD değeri de sınırlar içerisinde kalan piksellerin kesin referans değerine göre farklılaştığı için değeri azalmaktadır. EBH değeri hata parametresi olduğundan bölütleme performansı düştüğünde artmaktadır. Üç gürültü tipinde de süperpiksel sayısı arttıkça performansı artarken gürültü seviyesi arttıkça performansının azaldığı sonucuna varılmıştır.



Şekil 5.10. ERC algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



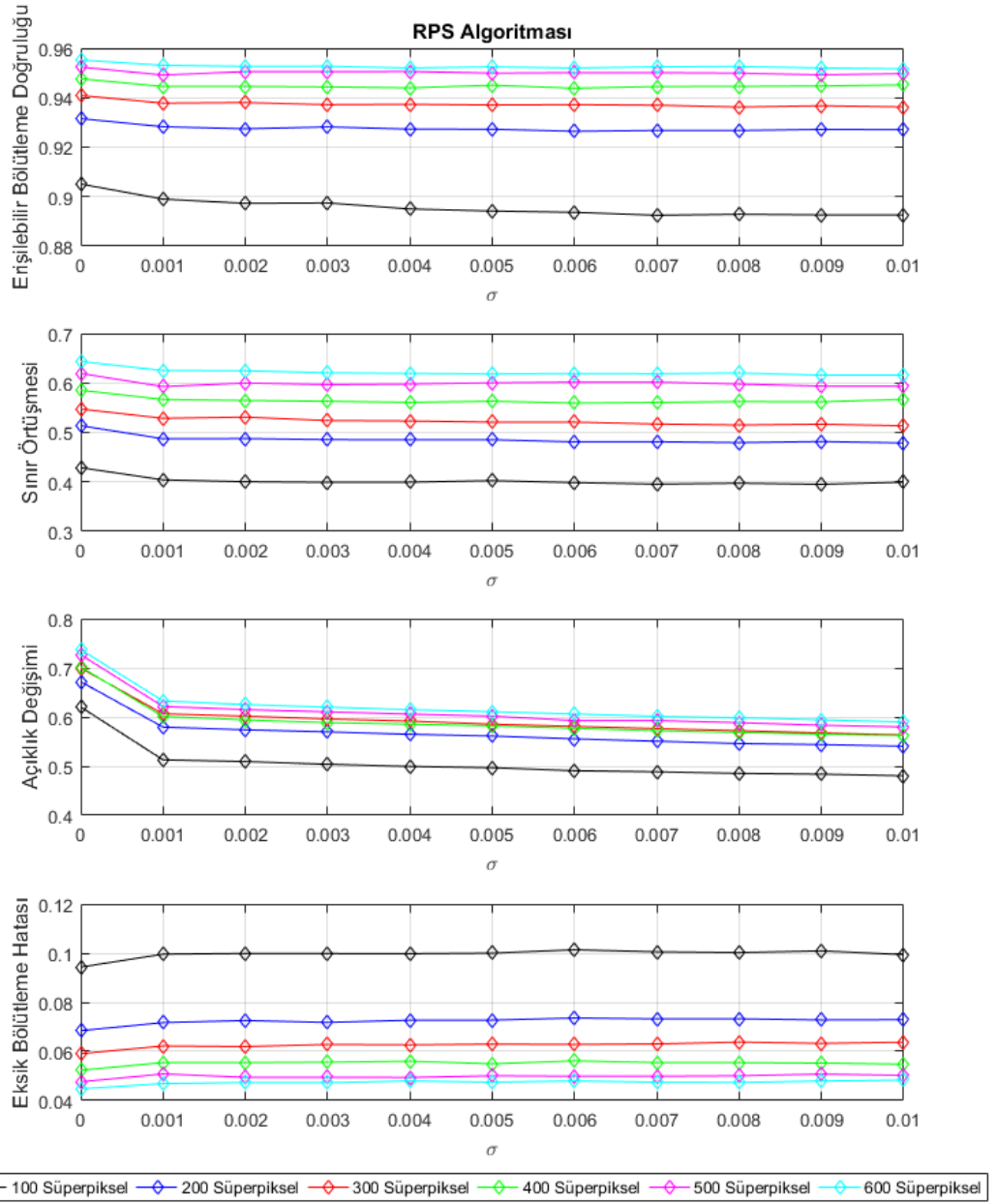
Şekil 5.11. ERC algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



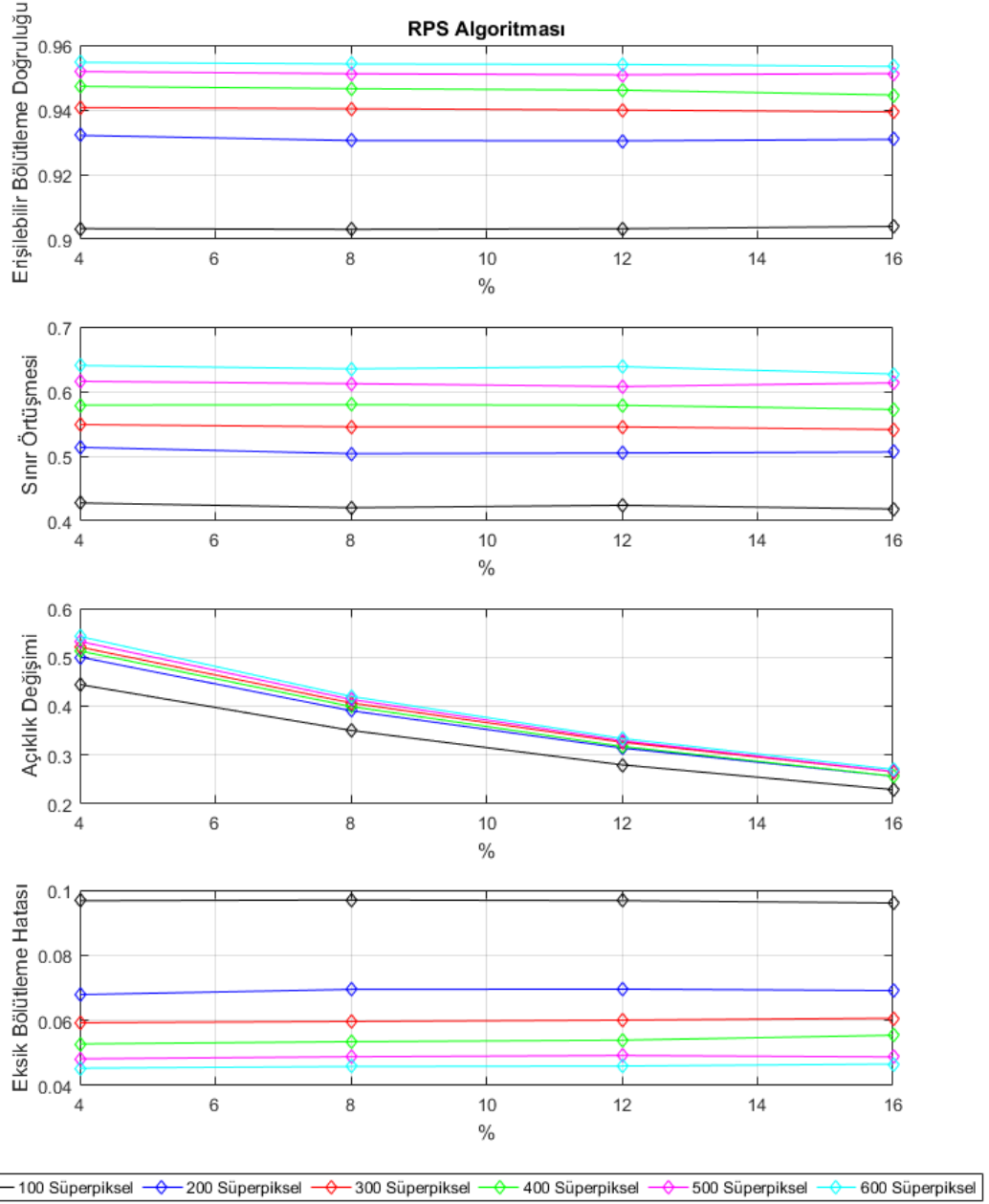
Şekil 5.12. ERC algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi

Şekil 5.13-5.15'te verilen grafikler RPS algoritmasının gürültü bağışıklığının incelenmesi için başarı metriklerinin sonuçlarını içermektedir. Şekil 5.13 ve Şekil 5.14'te sırasıyla gauss ve tuz&biber gürültüsünün RPS algoritmasına etkisi incelenmiştir. Burada eklenen gürültüler orijinal imgenin içeriğini bozduğu gözlemlenmiştir ve başarı metrikleri oldukça düşmektedir. Şekil 5.15'te RPS algoritmasının bulanıklaştırma etkisi incelenmiştir. Bulanıklaştırma etkisi arttıkça imge piksel değerleri birbirine yaklaşmaktadır. Bu yüzden AD değeri

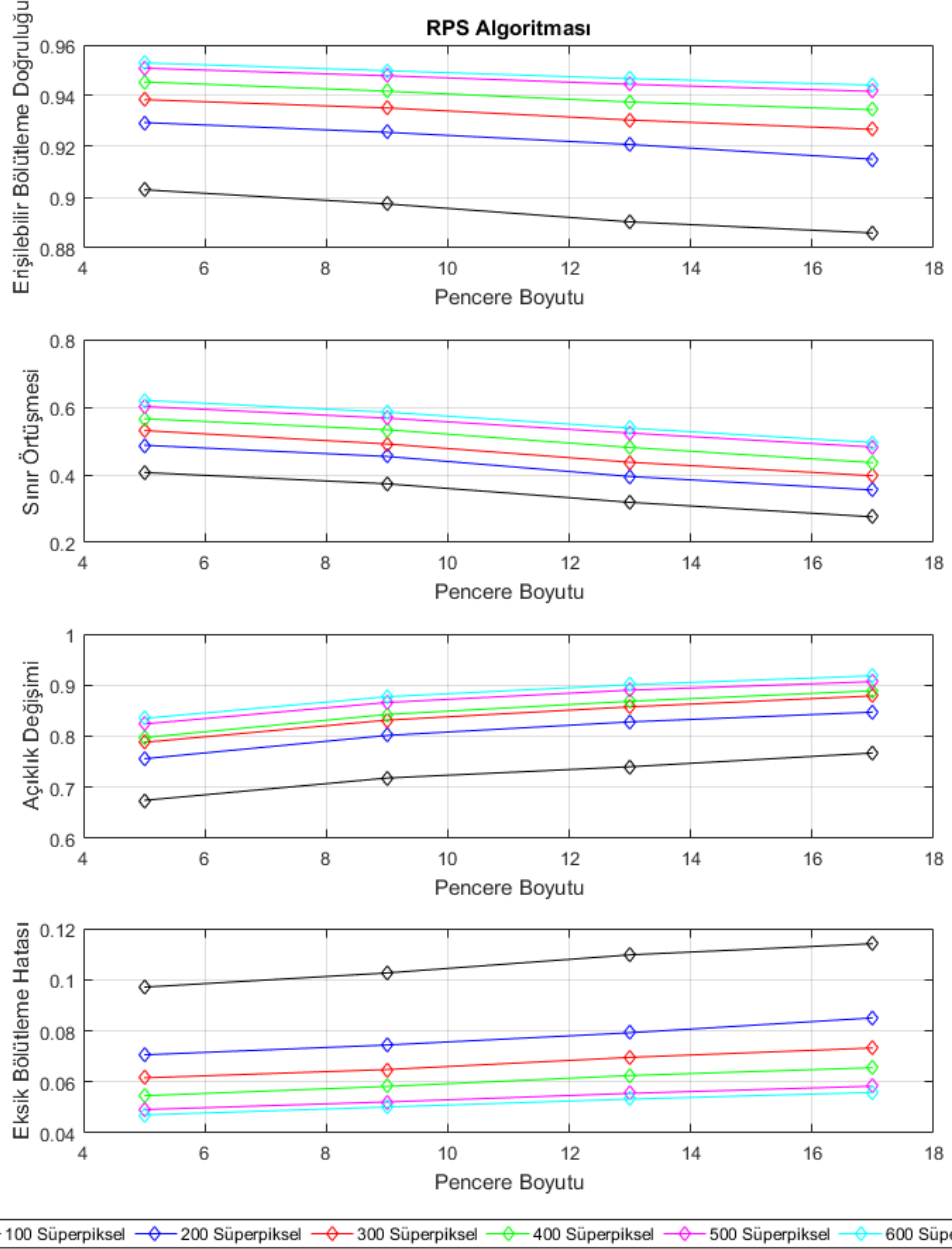
süperpikselerin homojenliğini incelediği için artmaktadır. SÖ değeri sınır değerlerindeki belirginlik azaldığı için azalmaktadır. EBD değeri de sınırlar içerisinde kalan piksellerin kesin referans değerine göre farklılaştığı için değeri azalmaktadır. EBH değeri hata parametresi olduğundan bölütleme performansı düştüğünde artmaktadır. Üç gürültü tipinde de süperpiksel sayısı arttıkça performansı artarken gürültü seviyesi arttıkça performansının azaldığı sonucuna varılmıştır.



Şekil 5.13. RPS algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



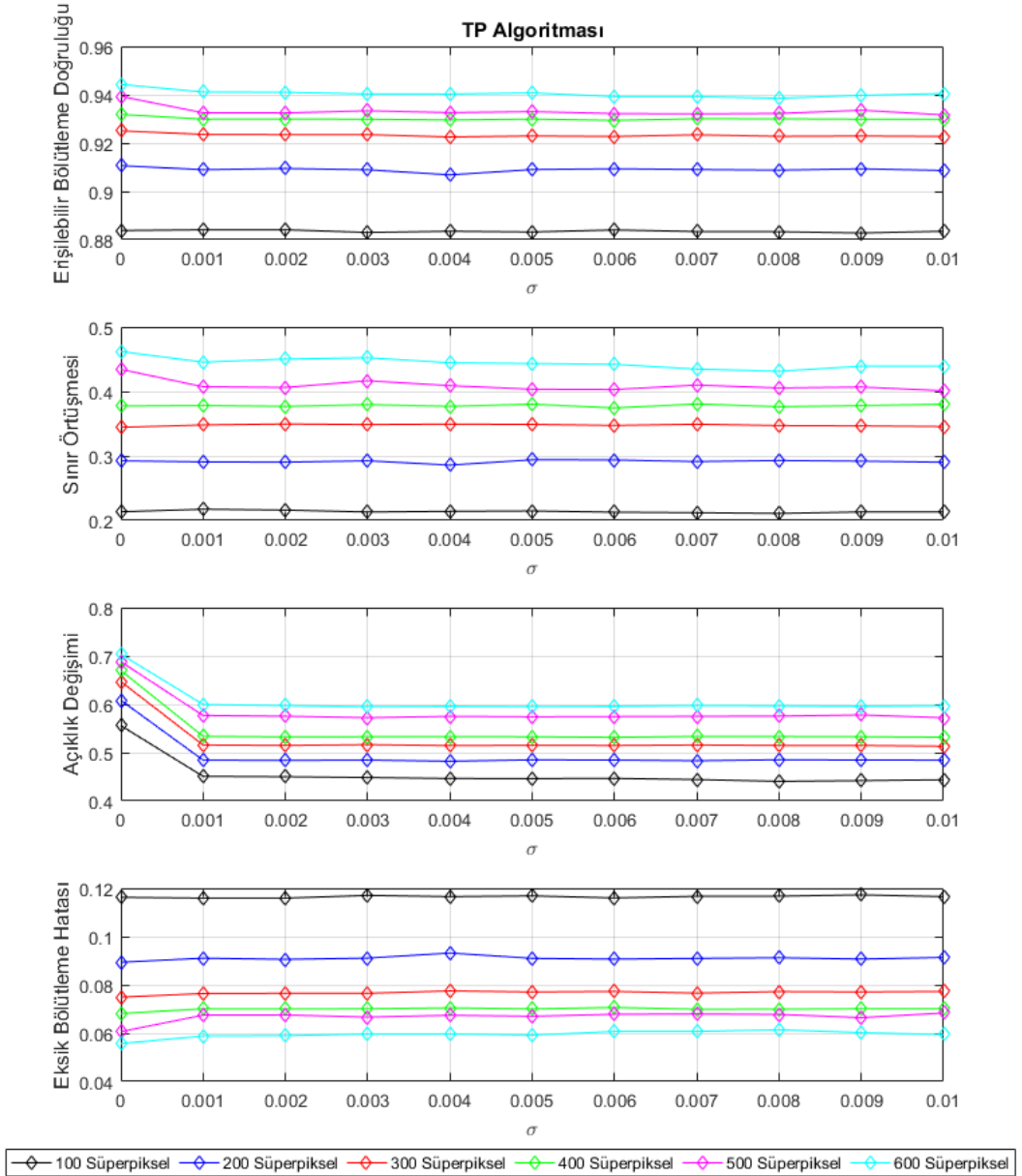
Şekil 5.14. RPS algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



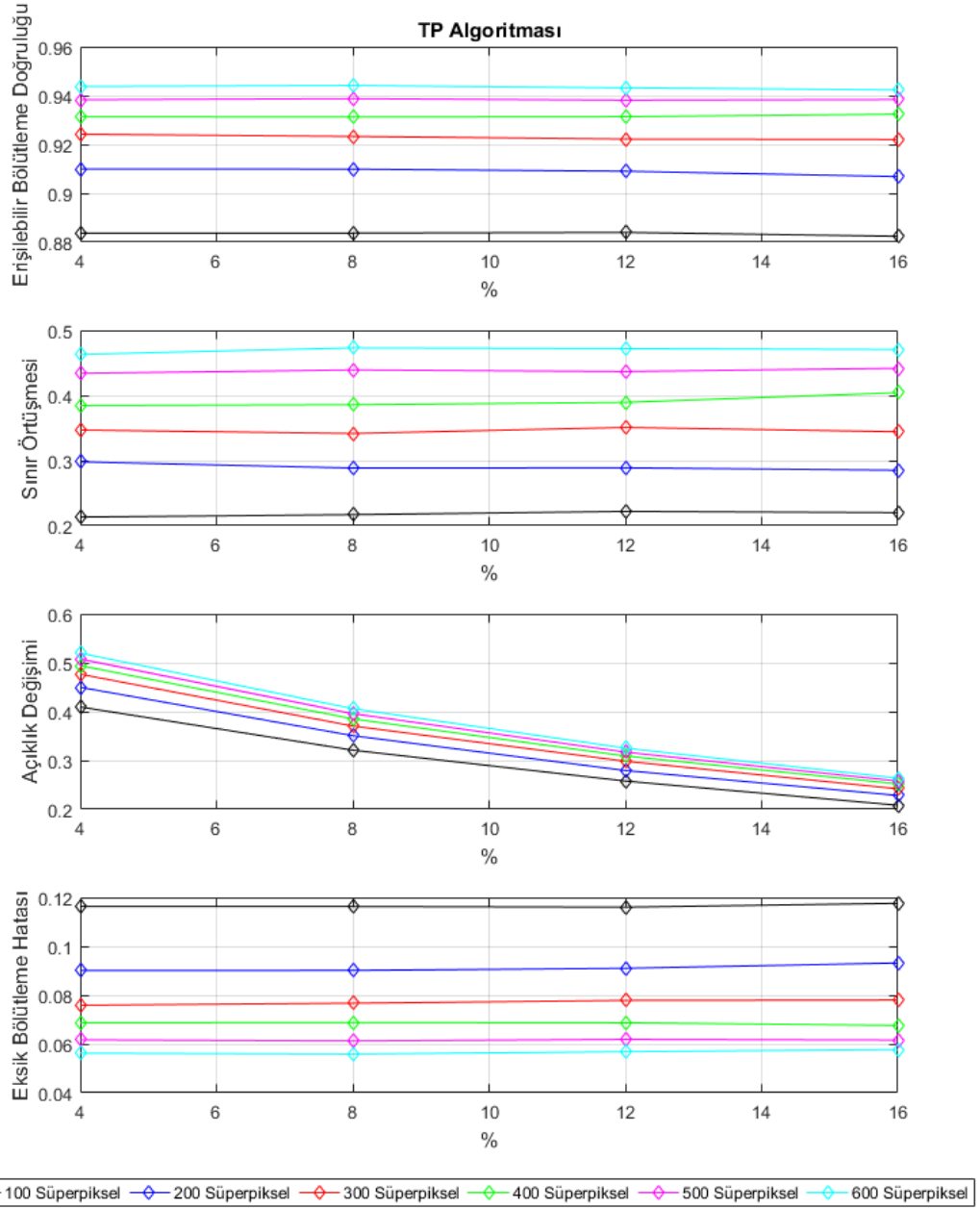
Şekil 5.15. RPS algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi

Şekil 5.16-5.18'de verilen grafikler TP algoritmasının gürültü bağımsızlığının incelenmesi için başarı metriklerinin sonuçlarını içermektedir. Şekil 5.16 ve Şekil 5.17'de sırasıyla gauss ve tuz&biber gürültüsünün TP algoritmasına etkisi incelenmiştir. Burada eklenen gürültüler orijinal imgenin içeriğini bozduğu gözlemlenmiştir ve başarı metrikleri oldukça düşmektedir. Şekil 5.18'de TP algoritmasının bulanıklaştırma etkisi incelenmiştir. Bulanıklaştırma etkisi arttıkça imge piksel değerleri birbirine yaklaşmaktadır. Bu yüzden AD değeri

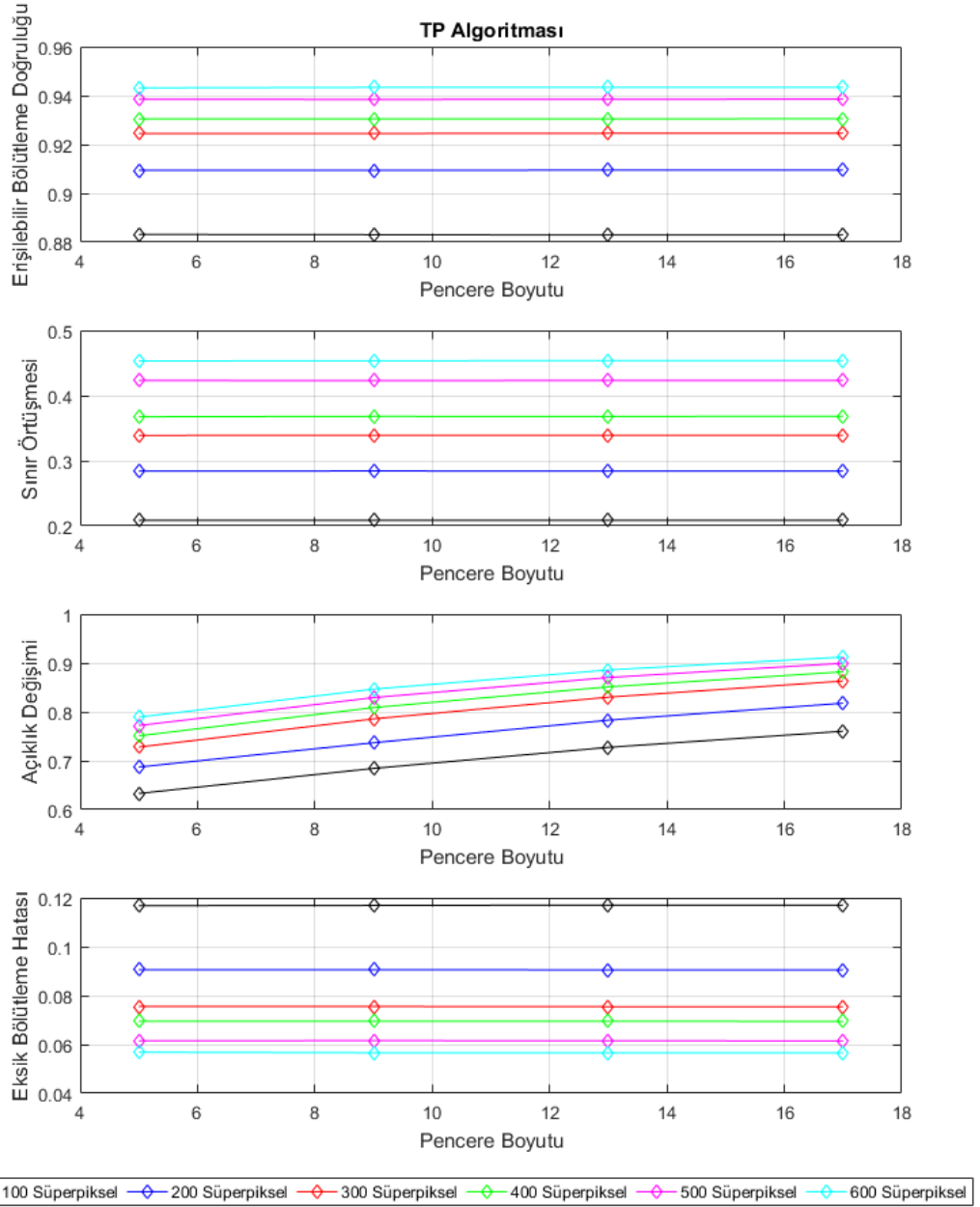
süperpikselerin homojenliğini incelediği için artmaktadır. SÖ değeri sınır değerlerindeki belirginlik azaldığı için azalmaktadır. EBD değeri de sınırlar içerisinde kalan piksellerin kesin referans değerine göre farklılaştığı için değeri azalmaktadır. EBH değeri hata parametresi olduğundan bölütleme performansı düştüğünde artmaktadır. Üç gürültü tipinde de süperpiksel sayısı arttıkça performansı artarken gürültü seviyesi arttıkça performansının azaldığı sonucuna varılmıştır.



Şekil 5.16. TP algoritmasının gauss gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



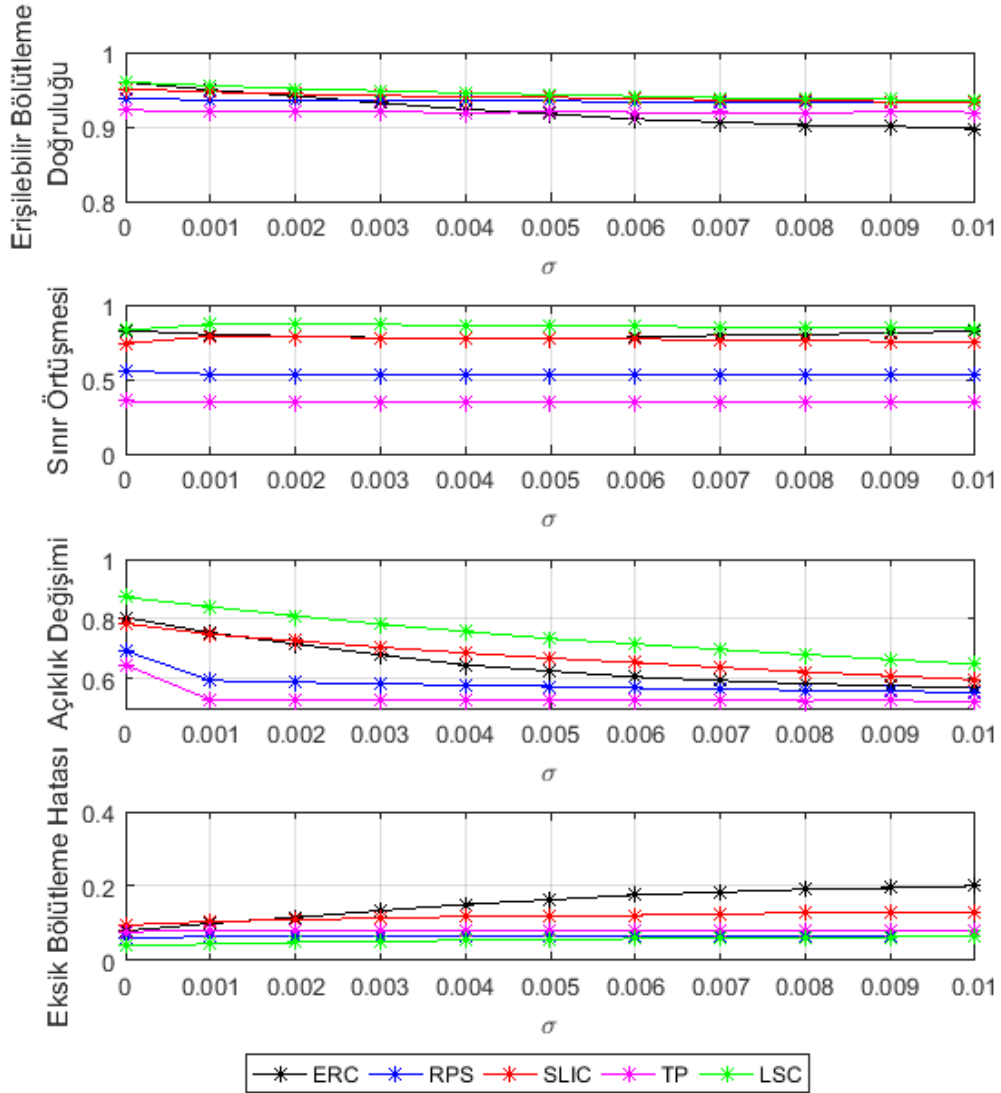
Şekil 5.17. TP algoritmasının tuz&biber gürültüsüne bağlı olarak metrik değerlerinin incelenmesi



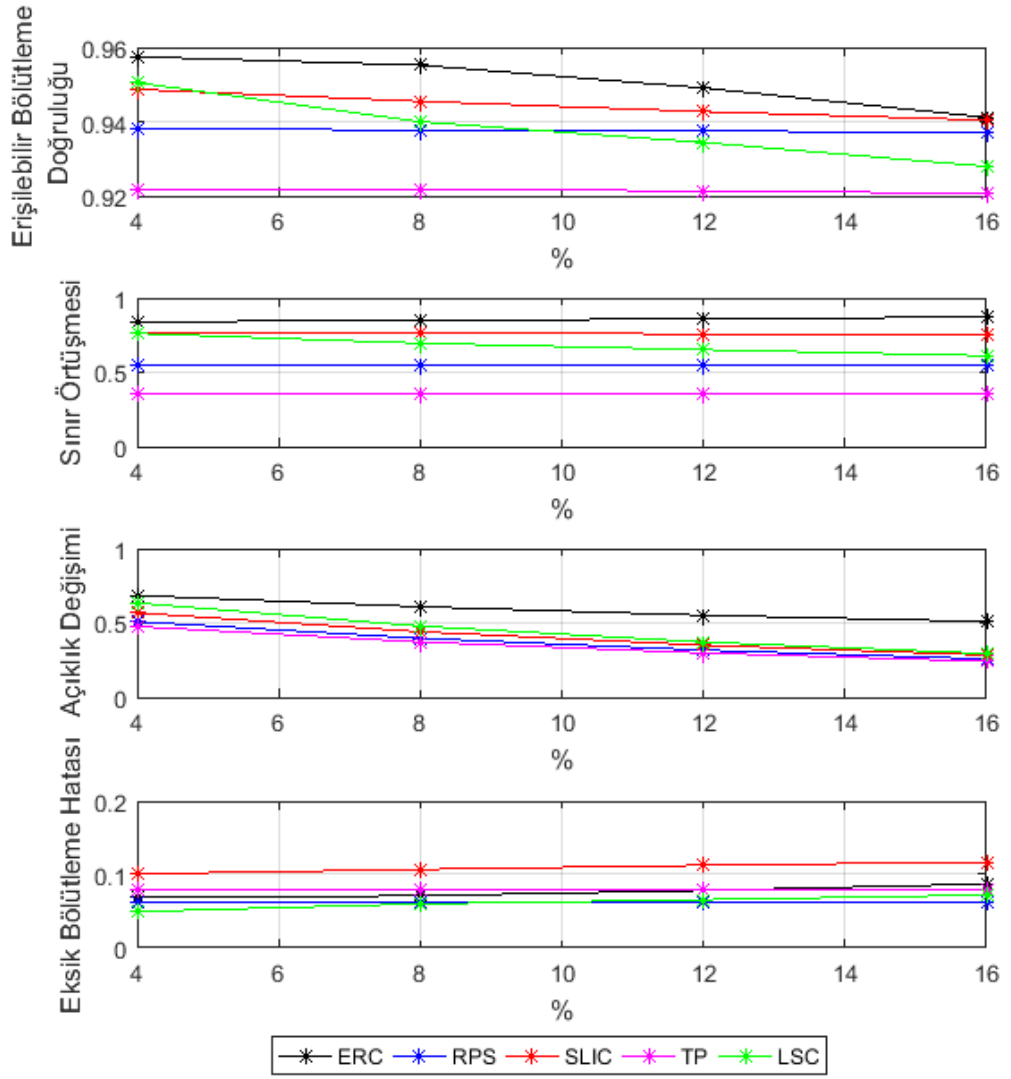
Şekil 5.18. TP algoritmasının bulanıklaştırma etkisine bağlı olarak metrik değerlerinin incelenmesi

Şekil 5.19-5.21 arasındaki şekillerde ortalama süperpiksel sayısı ile tez kapsamında kullanılan 5 algoritmanın performanslarını değerlendirebilmek için başarı metriklerinin gürültü bağımsızlığı incelenmiştir. Şekil 5.19’da gauss gürültüsünün bağımsızlığı incelenmiştir. Başarı metrikleri incelendiğinde en iyi performansa sahip algoritma LSC’dir. Gürültü bağımsızlığı incelendiğinde de LSC algoritması oldukça iyi bir performans sergilemektedir fakat TP algoritmasının performansının düşük olmasına rağmen gürültü bağımsızlığı açısından daha iyi sonuç vermektedir. Şekil 5.20’de tuz&biber gürültüsünün süperpiksel

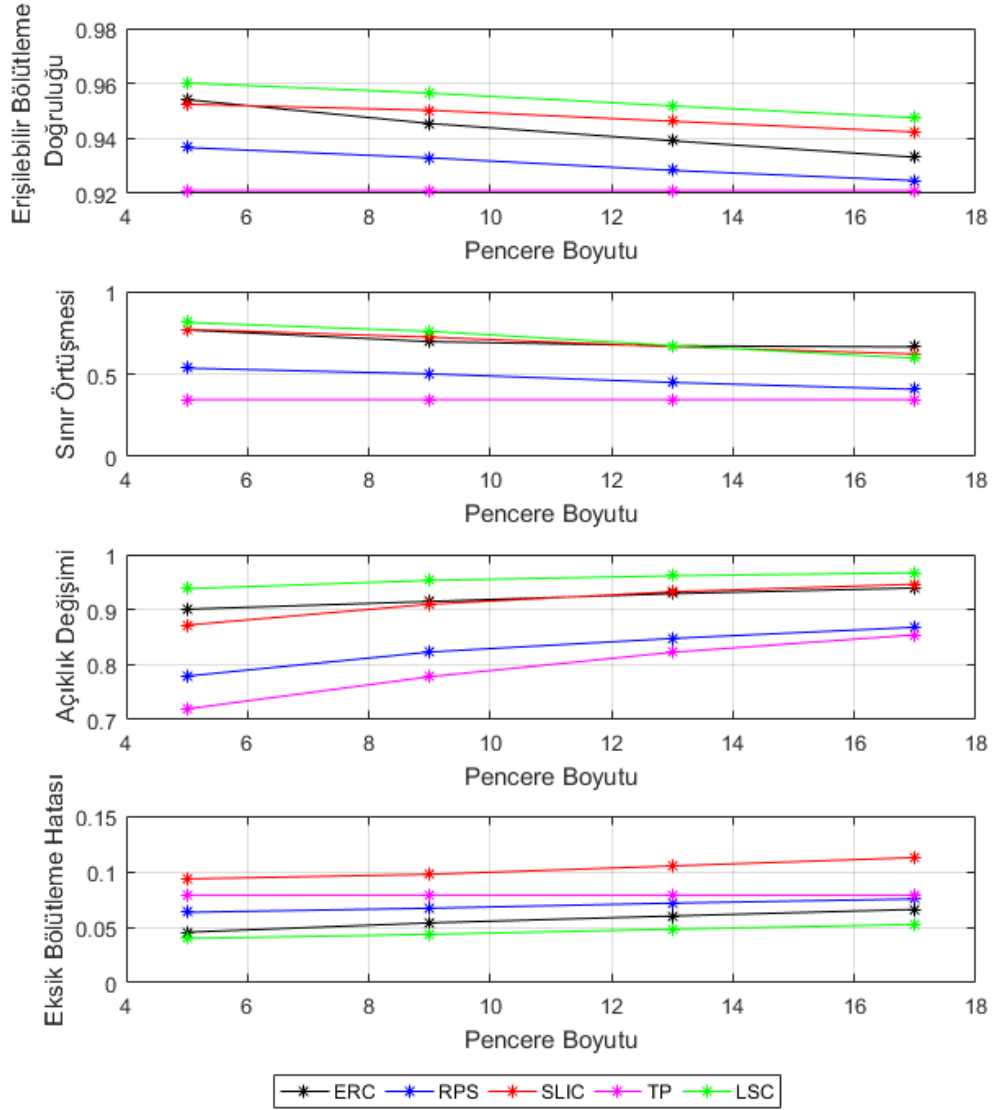
algoritmalarına etkisi incelenmektedir. Süperpiksel algoritmalarının performansı değerlendirildiğinde tuz&biber gürültülü imgeler üzerinde en iyi algoritmanın ERC algoritması olduğu açıkça görülmektedir. Fakat gürültü bağışıklığı oldukça düşüktür. Gürültü bağışıklığı açısından incelendiğinde TP ve RPS algoritması karşılaştırılabilir derecede iyi olduğu gözlemlenmektedir. Şekil 5.21’de bulanıklık etkisinin süperpiksel algoritması üzerindeki performansları verilmiştir. Şekil 5.16’da olduğu gibi en iyi performansı LSC algoritması sergilemektedir. Pencere boyutu arttıkça EBD, SÖ ve EBH metriklerinde performans değerleri azalmaktadır. Fakat pencere boyutu arttıkça AD metriği artmaktadır. Bunun nedeni komşuluğa bağlı olarak piksel değerlerinin birbirine yaklaşmasıyla süperpiksel içerisindeki homojenliğin artmasından kaynaklanmaktadır.



Şekil 5.19. Süperpiksel algoritmalarının gauss gürültüsüne bağlı değişimi

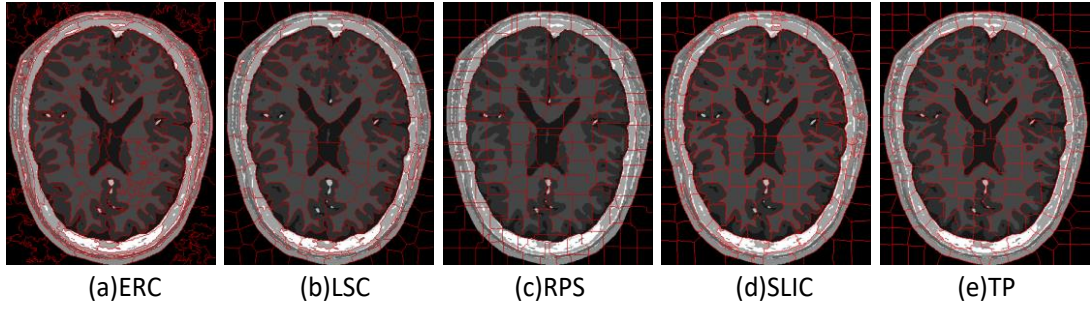


Şekil 5.20. Süperpiksel algoritmalarının tuz&biber gürültüsüne bağlı değişimi

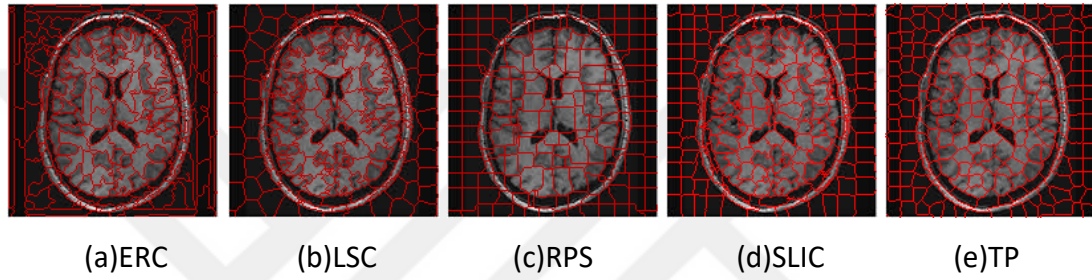


Şekil 5.21. Süperpiksel algoritmalarının bulanıklaştırma etkisine bağlı değişimi

Şekil 5.22 ve Şekil 5.23'te beyin MR görüntüleri için süperpiksel algoritmalarının sonuçları verilmiştir. Bu görüntüler BrainWeb veri kümesinden 20 normal beyin görüntüsünden 2 tanesi örnek olması amacıyla kullanılmıştır (http://brainweb.bic.mni.mcgill.ca/brainweb/anatomic_normal_20.html). Beyin MR görüntülerinde beyaz bölge, gri bölge, arka plan, kas vb. bölgelerin çıkarımı oldukça önemlidir. Bu bölgelerin kenar uyumluluğunda en iyi sonucu LSC algoritması sunmuştur. En kötü sonucu veren algoritma ise RPS algoritmasıdır. Çünkü RPS algoritması süperpikselleri oluştururken düzenlilik kriterini ön planda tutmaktadır. Böylece oluşan süperpikseller kenarları tespit etmekte başarı yakalayamamaktadır.



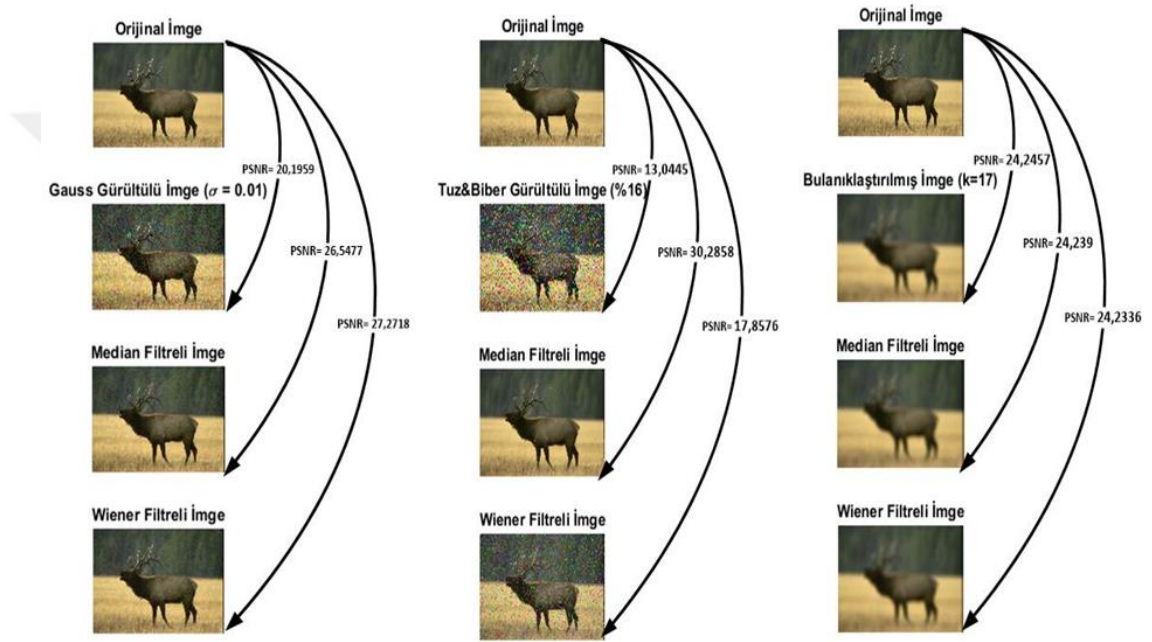
Şekil 5.22. Beyin MR görüntüsünün süperpiksel algoritmaları için sonuçları-1



Şekil 5.23. Beyin MR görüntüsünün süperpiksel algoritmaları için sonuçları-2

Çalışmanın üçüncü aşaması olan gürültülü imgelere filtre uygulanarak orijinal imgeye ne kadar benzediği ve filtreli imgelerin süperpiksel performanslarının incelenmesi için öncelikle *PSNR* başarı metriği değerlendirilmiştir. Burada *PSNR* değeri orijinal imge ile gürültülü imge arasında, orijinal imge ile medyan filtreli imge ve orijinal imge ile wiener filtreli imge arasında hesaplanmaktadır. Şekil 5.24'te gauss gürültüsü, tuz&biber gürültüsü ve bulanıklaştırılmış imgeler için bu çalışmada kullanılan en üst seviyede gürültü eklenerek oluşturulmuştur. Bu gürültülü imgelere medyan ve wiener filtre uygulanarak *PSNR* değerleri ölçülmüştür. Burada gauss gürültülü imgeye uygulanan medyan ve wiener filtre karşılaştırıldığında *PSNR* değeri yüksek olan wiener filtredir yani imgenin orijinal imgeye daha çok yaklaştığı gözlemlenmektedir. Tuz&biber gürültüsüne uygulanan filtrelerden medyan filtrenin *PSNR* değeri daha yüksektir yani tuz&biber gürültülü imge için medyan filtre orijinal imgeye daha da benzediği sonucuna varılmaktadır. Son olarak bulanıklaştırılmış imge için uygulanan filtreler performans açısından orijinal imgeye benzeme açısından iyi bir *PSNR*

sonucu vermemektedir. Kısaca bulanıklaştırılmış imgede bir nevi filtrelenmiş imge gibi algılandığı için performansı filtreler iyileştirmektedir. Ayrıca Çizelge 5.1'de tüm imgelerin orijinal imge ile gürültülü imge arasında, orijinal imge ile medyan filtreli imge ve orijinal imge ile wiener filtreli imge arasında tez çalışmasında kullanılan 15 test imgesi için ortalama *PSNR* değeri verilmiştir. Şekil 5.24'te olduğu gibi Çizelge 5.1'de de gauss gürültülü imge için wiener filtre, tuz&biber gürültülü imge için medyan filtre arasında ölçülen *PSNR* değeri ile orijinal imgeye daha da benzediği yani iyileştirildiği sonucuna varılmıştır.



Şekil 5.24. Gürültülü imgelerin filtreler ile iyileştirme performansının incelenmesi

Çizelge 5.1. Gürültülü imgelere uygulanan filtrelerin ortalama *PSNR* değeri

	Gürültülü İmge	Medyan Filtreli İmge	Wiener Filtreli İmge
Gauss gürültüsü (0.01)	20,36596	26,33111	27,03686
Tuz&Biber gürültüsü (% 16)	12,86845	29,64084	17,60111
Bulanıklık (17)	25,14131	25,12298	25,13261

Çizelge 5.2-5.5 arasındaki çizelgelerde orijinal imge, gürültülü imge ve filtreli imgelerin sırasıyla SÖ, EBD, AD ve EBH başarı metrikleri süperpiksel

algoritmaları için başarı metrikleri verilmiştir. Çizelge 5.2’de sınır örtüşmesi başarı metriğinin orijinal, gürültülü ve filtreli imgeler için süperpiksel algoritmalarının sonuçları verilmiştir. Burada en iyi performansı LSC algoritması ve ERC algoritması sunmaktadır. En kötü performansı ise TP algoritması sunmaktadır. Fakat gürültü bağışıklığı en yüksek algoritma TP algoritmasıdır.

Çizelge 5.2. Filtreli imgelerin SÖ başarı metriğinin incelenmesi

Gürültü Tipi	SÖ					
	Filtre	RPS	SLIC	TP	ERC	LSC
Gauss	Filtresiz	0,4994	0,7392	0,2743	0,7294	0,8239
	Medyan	0.5155	0.7248	0.2724	0,7506	0,8119
	Wiener	0.5219	0.7356	0.2722	0,7782	0,8202
Tuz & Biber	Filtresiz	0,5481	0,7600	0,2852	0,8704	0,6245
	Medyan	0.5323	0.7513	0.2736	0,8189	0,8056
	Wiener	0.5093	0.7164	0.2722	0,8189	0,7399
Bulanıklık	Filtresiz	0,4276	0,6686	0,2722	0,7135	0,6514
	Medyan	0.4246	0.6649	0.2722	0,7009	0,7183
	Wiener	0.4198	0.6606	0.2722	0,6966	0,7186
Orijinal		0.5398	0.7352	0.2797	0,8433	0,8304

Çizelge 5.3’te süperpiksel algoritmaları için erişilebilir bölütleme doğruluğu başarı metriği verilmiştir. Burada yine en iyi performansı LSC algoritması sunmaktadır.

Çizelge 5.3. Filtreli imgelerin EBD başarı metriğinin incelenmesi

Gürültü Tipi	EBD					
	Filtre	RPS	SLIC	TP	ERC	LSC
Gauss	Filtresiz	0,9339	0,9378	0,9097	0,9117	0,9428
	Medyan	0.9366	0.9507	0.9120	0,9568	0,9617
	Wiener	0.9371	0.9538	0.9126	0,9598	0,9619
Tuz & Biber	Filtresiz	0,9406	0,9521	0,9127	0,9522	0,9381
	Medyan	0.9386	0.9596	0.9127	0,9689	0,9703
	Wiener	0.9370	0.9469	0.9126	0,9566	0,9542
Bulanıklık	Filtresiz	0,9278	0,9581	0,9126	0,9458	0,9603
	Medyan	0.9274	0.9583	0.9126	0,9518	0,9655
	Wiener	0.9269	0.9584	0.9126	0,9519	0,9654
Orijinal		0.9391	0.9579	0.9132	0,9705	0,9682

Çizelge 5.4'te süperpiksel algoritmalarının orijinal, gürültülü ve filtreli imgeler için açıklık değişimi başarı metriğinin sonuçları yer almaktadır. Algoritmalar karşılaştırıldığında en iyi performansı LSC algoritması ve en kötü performansı TP algoritması sunmaktadır. Orijinal imge ile bulanıklaştırılmış imgeler karşılaştırıldığında tüm algoritmalarda bulanıklaştırılmış imgelerin AD değerinin daha iyi olduğu açıkça gözlemlenmektedir.

Çizelge 5.4. Filtreli imgelerin AD başarı metriğinin incelenmesi

Gürültü Tipi	AD					
	Filtre	RPS	SLIC	TP	ERC	LSC
Gauss	Filtresiz	0,6856	0,7550	0,6252	0,7646	0,7292
	Medyan	0.8166	0.8648	0.7618	0,8996	0,9175
	Wiener	0.8286	0.8815	0.7751	0,8950	0,9092
Tuz & Biber	Filtresiz	0,4692	0,4990	0,4370	0,6725	0,4262
	Medyan	0.8302	0.8885	0.7743	0,9348	0,9450
	Wiener	0.7327	0.7714	0.6845	0,7350	0,7393
Bulanıklık	Filtresiz	0,8847	0,9416	0,8326	0,9319	0,9667
	Medyan	0.8843	0.9424	0.8464	0,9527	0,9696
	Wiener	0.8869	0.9449	0.8473	0,9532	0,9696
Orijinal		0.8115	0.8722	0.7604	0,8871	0,9281

Çizelge 5.5'te algoritmaların orijinal, gürültülü ve filtreli imgeler için eksik bölütleme hatası başarı metriği sonuçları verilmiştir. Algoritmalar karşılaştırıldığında en iyi performansı LSC algoritması ve en kötü performansı SLIC algoritması sunmaktadır.

Çizelge 5.5. Filtreli imgelerin EBH başarı metriğinin incelenmesi

Gürültü Tipi	EBH					
	Filtre	RPS	SLIC	TP	ERC	LSC
Gauss	Filtresiz	0,06603	0,1221	0,0902	0,1731	0,0571
	Medyan	0.0634	0.0971	0.0880	0,0853	0,0382
	Wiener	0.0629	0.0908	0.0874	0,0796	0,0380
Tuz & Biber	Filtresiz	0,0593	0,0941	0,0872	0,0720	0,0618
	Medyan	0.0614	0.0795	0.0873	0,0572	0,0296
	Wiener	0.0630	0.1043	0.0874	0,0813	0,0457
Bulanıklık	Filtresiz	0,0721	0,0817	0,0873	0,0672	0,0396
	Medyan	0.0726	0.0813	0.0874	0,0906	0,0344
	Wiener	0.0731	0.0814	0.0874	0,0904	0,0345
Orijinal		0.0609	0.0829	0.0868	0,0582	0,0317

6. TARTIŞMA

Literatürde bilgisayarlı görü sistemlerinde, medikal imge bölütleme çalışmalarında nesne tespiti, sınır tespiti gibi uygulamalarda süperpiksel algoritmaları sıklıkla kullanılmaktadır. Tez kapsamında bölütleme için en yaygın kullanılan Berkeley veri kümesinden farklı renk, doku ve yoğunluk değerine sahip 15 imge seçilmiştir. 15 test imgesine 3 farklı gürültü tipi ve 2 farklı filtre tipi uygulanarak toplam 855 imge üzerinde işlem yapılmıştır. Tez çalışmasında kullanılan 855 imgeye literatürde en çok kullanılan SLIC, LSC, RPS, ERC ve TP olmak üzere 5 farklı süperpiksel algoritması uygulanmıştır. Süperpiksel algoritmaları 100-600 süperpiksel sayısı için 6 farklı süperpiksel sayısı üzerinden işlem yapılmıştır.

Bu tez kapsamında literatürde en çok kullanılan süperpiksel algoritmalarından SLIC, RPS, LSC, ERC ve TP süperpiksel algoritmaları kullanılmıştır. Kullanılan bu algoritmalar üç farklı aşama ile imgelere uygulanmıştır. İlk aşama veri kümesinden seçilen 15 test imgesine orijinal şeklinde uygulanmıştır. İkinci aşama ise veri iletimi sırasında maruz kalınan gürültü nedeniyle ve medikal cihazlardan alınan imgelerin gürültü içermesi nedeniyle süperpiksel algoritmalarının gürültülü imgeler üzerindeki performansını değerlendirebilmek amacıyla orijinal imgelere gauss gürültüsü, tuz&biber gürültüsü ve bulanıklaştırma etkisi eklenmiştir. Oluşturulan bu gürültülü sentetik imgelere süperpiksel algoritmaları uygulanmıştır. Üçüncü aşama ise gürültülü imgelerin performansı filtre uygulanarak imgeler iyileştirilip süperpiksel algoritmalarının başarı metrikleri karşılaştırılmıştır. Tez çalışmasında orijinal imgelere eklenen gürültü performansları iyileştirmek için literatürdeki en iyi sonucu veren filtreler kullanılmıştır. Fakat haberleşme sistemin düşündüğümüzde hangi gürültüye maruz kaldığı bilinmediği için eklenen gürültülere medyan ve wiener filtreler uygulanmıştır. Böylece filtre uygulanarak süperpiksel algoritmalarının başarıları karşılaştırılmıştır.

Orijinal imgelere uygulanan süperpiksel algoritmalarından en başarılı performansı sergileyen süperpiksel algoritması LSC algoritmasıdır. Orijinal imgelere eklenen

gürültü ile gürültülü imgeler oluşturulmuştur. Gauss gürültülü ve bulanıklaştırılmış imgelere uygulanan süperpiksel algoritmalarının performansı karşılaştırıldığında orijinal imgede olduğu gibi LSC süperpiksel algoritması en başarılı algoritmadır. Fakat tuz&biber gürültülü imgelere uygulanan süperpiksel algoritmalarından ERC algoritması en başarılı performansı sergilemektedir. Farklı gürültü tiplerine uygulanan filtrelerin aynı iyileştirme performansı sergilemediği *PSNR* değeri ile incelenmiştir. Burada gauss gürültülü imgeyi iyileştirmek için wiener filtre daha iyi *PSNR* değeri sunarken, tuz&biber gürültülü imgeyi iyileştirmek için medyan filtre daha iyi *PSNR* değeri sunmaktadır. Bulanıklaştırılmış imgeye uygulanan medyan ve wiener filtrenin performansı değiştirmedeği gözlemlenmiştir. Bu sonuçlarla imgeleri iyileştirebilmek için gürültü tipinin ve ona uygun filtrenin kullanılması gerekmektedir. Bu sonuçlarla orijinal imge, gürültülü imge ve filtreli imgelere tez kapsamında kullanılan süperpiksel algoritmaları uygulanmıştır ve elde edilen başarı metriklerinin değerlerinden yola çıkılarak bu algoritmalarından en iyi sonucu LSC süperpiksel algoritması sunmuştur.

7. KAYNAKLAR

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11), 2274-2282.
- Benesova, W., & Kottman, M. (2014). Fast superpixel segmentation using morphological processing. In *Conference on Machine Vision and Machine Learning*.
- Buyskens, P., Gardin, I., Ruan, S., & Elmoataz, A. (2014). Eikonal-based region growing for efficient clustering. *Image and Vision Computing*, 32(12), 1045-1054.
- Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on image processing*, 10(2), 266-277.
- Chen, J., Li, Z., & Huang, B. (2017). Linear spectral clustering superpixel. *IEEE Transactions on Image Processing*, 26(7), 3317-3330.
- Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5), 603-619.
- Despotović, I., Jelača, V., Vansteenkiste, E., & Philips, W. (2010). Noise-robust method for image segmentation. In *Advanced Concepts for Intelligent Vision Systems* (pp. 153-162). Springer Berlin/Heidelberg.
- Ertürk, A. (2016, May). Denoising in hyperspectral images by superpixel based unmixing. In *Signal Processing and Communication Application Conference (SIU), 2016 24th* (pp. 2189-2192). IEEE.
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2), 167-181.
- Fu, H., Cao, X., Tang, D., Han, Y., & Xu, D. (2014). Regularity preserved superpixels and supervoxels. *IEEE Transactions on Multimedia*, 16(4), 1165-1175.
- Fu, P., Li, C., Sun, Q., Cai, W., & Feng, D. D. (2014, October). Image noise level estimation based on a new adaptive superpixel classification. In *Image Processing (ICIP), 2014 IEEE International Conference on* (pp. 2649-2653). IEEE.
- Gonzalez, R. C., & Woods, R. E. (1992). *Digital image processing*.
- Hu, Z., Zou, Q., & Li, Q. (2015, September). Watershed superpixel. In *Image Processing (ICIP), 2015 IEEE International Conference on* (pp. 349-353). IEEE.

- Ji, S., Wei, B., Yu, Z., Yang, G., & Yin, Y. (2014). A new multistage medical segmentation method based on superpixel and fuzzy clustering. *Computational and mathematical methods in medicine*.
- Levinshtein, A., Sminchisescu, C., & Dickinson, S. (2010). Optimal contour closure by superpixel grouping. *Computer Vision–ECCV 2010*, 480-493.
- Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., & Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12), 2290-2297.
- Li, Z., Wu, X. M., & Chang, S. F. (2012, June). Segmentation using superpixels: A bipartite graph partitioning approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 789-796). IEEE.
- Liu, B., Hu, H., Wang, H., Wang, K., Liu, X., & Yu, W. (2013). Superpixel-based classification with an adaptive number of classes for polarimetric SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2), 907-924.
- Liu, M. Y., Tuzel, O., Ramalingam, S., & Chellappa, R. (2011, June). Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 2097-2104). IEEE.
- Liu, M. Y., Tuzel, O., Ramalingam, S., & Chellappa, R. (2014). Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1), 99-112.
- Machairas, V., Faessel, M., Cárdenas-Peña, D., Chabardes, T., Walter, T., & Decencière, E. (2015). Waterpixels. *IEEE Transactions on Image Processing*, 24(11), 3707-3716.
- Makni, N., Puech, P., Lopes, R., Dewalle, A. S., Colot, O., & Betrouni, N. (2009). Combining a deformable model and a probabilistic framework for an automatic 3D segmentation of prostate on MRI. *International journal of computer assisted radiology and surgery*, 4(2), 181-188.
- Moore, A. P., Prince, S. J., Warrell, J., Mohammed, U., & Jones, G. (2008, June). Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.
- Mori, G. (2005, October). Guiding model search using segmentation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Vol. 2, pp. 1417-1423). IEEE.
- Mori, G., Ren, X., Efros, A. A., & Malik, J. (2004, June). Recovering human body configurations: Combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (Vol. 2, pp. II-II). IEEE.

- Özer F, Özkaya U (2017). "Süperpiksel Algoritmalarının Gürültülü İmgeler için Bölütleme Performanslarının İncelenmesi", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU).
- Peng, B., Zhang, L., & Zhang, D. (2011). Automatic image segmentation by dynamic region merging. *IEEE Transactions on image processing*, 20(12), 3592-3605.
- Peng, J., Shen, J., Yao, A., & Li, X. (2016). Superpixel Optimization Using Higher Order Energy. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(5), 917-927.
- Ren, X., & Malik, J. (2003, October). Learning a classification model for segmentation. In null (p. 10). IEEE.
- Roels, J., De Vylder, J., Aelterman, J., Lippens, S., Saeys, Y., & Philips, W. (2016, March). Superpixel quality in microscopy images: the impact of noise & denoising. In *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016: MEDICON 2016*, March 31st-April 2nd 2016, Paphos, Cyprus (Vol. 57, p. 258). Springer.
- Shen, J., Du, Y., Wang, W., & Li, X. (2014). Lazy random walks for superpixel segmentation. *IEEE Transactions on Image Processing*, 23(4), 1451-1462.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), 888-905.
- Stutz, D., Hermans, A., & Leibe, B. (2017). Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*.
- Sun, X., Shang, K., Ming, D., Tian, J., & Ma, J. (2015). A biologically-inspired framework for contour detection using superpixel-based candidates and hierarchical visual cues. *Sensors*, 15(10), 26654-26674.
- Tang, D., Fu, H., & Cao, X. (2012, July). Topology preserved regular superpixel. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on* (pp. 765-768). IEEE.
- Tasli, H. E., Cigla, C., & Alatan, A. A. (2015). Convexity constrained efficient superpixel and supervoxel extraction. *Signal Processing: Image Communication*, 33, 71-85.
- Tian, Z., Liu, L., Zhang, Z., & Fei, B. (2016). Superpixel-based segmentation for 3d prostate mr images. *IEEE transactions on medical imaging*, 35(3), 791-801.
- Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., & Van Gool, L. (2012, October). SEEDS: Superpixels extracted via energy-driven sampling. In *European conference on computer vision* (pp. 13-26). Springer, Berlin, Heidelberg.
- Vedaldi, A., & Soatto, S. (2008). Quick shift and kernel methods for mode seeking. *Computer vision–ECCV 2008*, 705-718.

- Veksler, O., Boykov, Y., & Mehrani, P. (2010). Superpixels and supervoxels in an energy optimization framework. *Computer Vision–ECCV 2010*, 211-224.
- Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 583-598.
- Wang, J., & Wang, X. (2012). VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations. *IEEE Transactions on pattern analysis and machine intelligence*, 34(6), 1241-1247.
- Wu, C. H., & Chang, H. H. (2015). Superpixel-based image noise variance estimation with local statistical assessment. *EURASIP Journal on Image and Video Processing*, 2015(1), 38.
- Yang, F., Lu, H., & Yang, M. H. (2014). Robust superpixel tracking. *IEEE Transactions on Image Processing*, 23(4), 1639-1651.
- Zhang, L., Gao, Y., Xia, Y., Lu, K., Shen, J., & Ji, R. (2014). Representative discovery of structure cues for weakly-supervised image segmentation. *IEEE Transactions on Multimedia*, 16(2), 470-479.
- Zhang, Y., Li, X., Gao, X., & Zhang, C. (2016). A simple algorithm of superpixel segmentation with boundary constraint. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Zhou, Y., Pan, X., Wang, W., Yin, Y., & Zhang, C. (2016). Superpixels by Bilateral Geodesic Distance. *IEEE Transactions on Circuits and Systems for Video Technology*.

ÖZGEÇMİŞ

Adı Soyadı : Fadime ÖZER
Doğum Yeri ve Yılı : Isparta, 1992
Medeni Hali : Bekar
Yabancı Dili : İngilizce
E-posta : fadimeozer.1992@gmail.com



Eğitim Durumu

Lise : Burdur Anadolu Lisesi, 2010
Lisans : SDÜ, Mühendislik Fakültesi, Elektronik ve Haberleşme Mühendisliği, 2014

Yayınları

Dursun G, Özer F, Özkaya U, A new and secure digital image scrambling algorithm based on 2D cellular automata, Turkish Journal Electrical Engineering & Computer Sciences, 2017, DOI: 10.3906/elk-1610-225

Özer F, Özkaya U, "Süperpiksel Algoritmalarının Gürültülü İmgeler için Bölütleme Performanslarının İncelenmesi", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU), 5-7 Ekim 2017, Alanya

Kaya B., Özer F., Dursun G., Aydoğdu G., Özkaya U., "Parçacık Sürü Optimizasyonu ve Hücresel Otomata Tabanlı Sayısal Görüntü Karıştırma Yöntemi", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU), 9-10 Ekim 2014, İzmir