

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YÜKSEK BOYUTLU MODEL GÖSTERİLİMİ ve
ÇOK DEĞİŞKENLİLİĞİ YÜKSELTİLMİŞ ÇARPIMLAR GÖSTERİLİMİ ile
GÖRÜNTÜ ÜZERİNDEKİ GÜRÜLTÜLERİ GİDERME**

DOKTORA TEZİ

Sena KAÇAR

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

EYLÜL 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YÜKSEK BOYUTLU MODEL GÖSTERİLİMİ ve
ÇOK DEĞİŞKENLİLİĞİ YÜKSELTİLMİŞ ÇARPIMLAR GÖSTERİLİMİ ile
GÖRÜNTÜ ÜZERİNDEKİ GÜRÜLTÜLERİ GİDERME**

DOKTORA TEZİ

**Sena KAÇAR
(509172206)**

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Burcu TUNGA

EYLÜL 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**IMAGE DENOISING via
HIGH DIMENSIONAL MODEL REPRESENTATION and
ENHANCED MULTIVARIATE PRODUCT REPRESENTATION**

Ph.D. THESIS

**Sena KAÇAR
(509172206)**

Department of Mathematical Engineering

Mathematical Engineering Programme

Thesis Advisor: Doç. Dr. Burcu TUNGA

SEPTEMBER 2024

İTÜ, Lisansüstü Eğitim Enstitüsü'nün 509172206 numaralı Doktora öğrencisi Sena KAÇAR, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “YÜKSEK BOYUTLU MODEL GÖSTERİLİMİ ve ÇOK DEĞİŞKENLİLİĞİ YÜKSELTİLMİŞ ÇARPIMLAR GÖSTERİLİMİ ile GÖRÜNTÜ ÜZERİNDEKİ GÜRÜLTÜLERİ GİDERME” başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Burcu TUNGA**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Süreyya AKYÜZ**
Bahçeşehir Üniversitesi

Prof. Dr. Behçet Uğur TÖREYİN
İstanbul Teknik Üniversitesi

Dr. Öğr. Üye. Evrim KORKMAZ ÖZAY
Beykent Üniversitesi

Dr. Öğr. Üye. Süha TUNA
İstanbul Teknik Üniversitesi

Teslim Tarihi : **26 Temmuz 2024**
Savunma Tarihi : **9 Eylül 2024**





Aileme,



ÖNSÖZ

Bu çalışmada Yüksek boyutlu Model Gösterilimi ve Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi kullanılarak görüntü üzerinde bulunan gürültüleri giderme işlemi yapılmıştır. Yapılan çalışma ile özellikle tuz ve biber gürültüsünü hem siyah beyaz görüntülerden, hem de renkli görüntülerden arındırılması amaçlanmıştır.

Doktora sürecim boyunca, yaşadığım tüm zorluklara ortak olan, bana emsalsiz motivasyon sağlayan ve hayatıma bir doktora tez danışmanının çok ötesinde varlığını katan sevgili hocam Doç. Dr. Burcu TUNGA'ya yürekten minnettarım. Bu zorlu süreçte beni sürekli destekleyen aileme ve özellikle eşime çok teşekkür ederim. Ayrıca zorlu ekonomik koşullar altında doktora eğitimimde beni destekleyen TÜBİTAK'a 'BİDEB 2211-A Yurt içi Genel Doktora Burs Programı' ile katkıda bulunduğu için teşekkür ederim.

Eylül 2024

Sena KAÇAR
Öğretim Görevlisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ	ix
İÇİNDEKİLER	xi
KISALTMALAR.....	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET	xix
SUMMARY	xxiii
1. GİRİŞ	1
1.1 Tezin Amacı ve Kapsamı.....	2
1.2 Özgün Değer ve Yaygın Etki.....	2
1.3 Literatür Özeti	3
2. KULLANILAN YÖNTEM	11
2.1 Yüksek Boyutlu Model Gösterilimi ile Fonksiyon Ayrıştırımı	11
2.2 Yüksek Boyutlu Model Gösterilimi ile Veri Ayrıştırımı	14
2.3 Yüksek Boyutlu Model Gösteriliminin Görüntüye Uygulanması.....	16
2.4 Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi ile Fonksiyon Ay- rıştırımı	18
2.5 Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi ile Veri Ayrıştırımı...	20
3. ÖNERİLEN YÖNTEM	23
4. DENEYSEL BULGULAR	37
5. SONUÇLAR.....	67
KAYNAKLAR.....	71
ÖZGEÇMİŞ	75



KISALTMALAR

ADY	: Ayrıntıya Duyarlı Filtre
AGMF	: Aşamalı Geçişli Medyan Filtre
ÇYÇG	: Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi
FUMF	: Farklı Uygulanabilir Medyan Filtre
KTA	: Karar Tabanlı Algoritma
KUIA	: Küresel Uyarlanabilir İntegral Algoritması
MF	: Medyan Filtre
MFUMF	: Modifiye Edilmiş Farklı Uygulanabilir Medyan Filtre
PYTF	: Piksel Yoğunluğu Tabanlı Filtre
SKIF	: Sürekli Kesirler İnterpolasyon Filtresi
UAOF	: Uyarlanabilir Ağırlıklı Ortalama Filtre
UMF	: Uyarlanabilir Medyan Filtre
UTTKA	: Uyarlanabilir Türev Tabanlı Genelleştirilmiş Kesirli Algoritma
YBMG	: Yüksek Boyutlu Model Gösterilimi
YKTA	: Yeni Karar Tabanlı Algoritma



ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1 : 0,01 varyanslı Gauss gürültülü görüntülerin PSNR değerleri.	38
Çizelge 4.2 : 0,05 varyanslı alacalı gürültülü görüntülerin PSNR değerleri.	39
Çizelge 4.3 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMGMF1 ve YBMGMF2 ile elde edilen PSNR değerleri.	40
Çizelge 4.4 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMGMF3 ve YBMGMF4 ile elde edilen PSNR değerleri.	40
Çizelge 4.5 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMGMF5 ile elde edilen PSNR değerleri.	41
Çizelge 4.6 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMG algoritmaları ile elde edilen PSNR değerleri.	43
Çizelge 4.7 : 0,1, 0,2 ve 0,3 yoğunluklu tuz ve biber gürültüsüne sahip Monarch görüntüsünün farklı filtrelerden elde edilen PSNR değerleri.	43
Çizelge 4.8 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.	44
Çizelge 4.9 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.	46
Çizelge 4.10 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.	47
Çizelge 4.11 : Monarch görüntüsünün farklı yoğunluklardaki farklı filtrelerle elde edilmiş PSNR ve SSIM değerleri.	47
Çizelge 4.12 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerden elde edilen PSNR ve SSIM değerleri.	49
Çizelge 4.13 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin YBMG ve ÇYCG filtreleri ile elde edilen PSNR ve SSIM değerleri.	50
Çizelge 4.14 : Kameraman görüntüsünün farklı yoğunluklardaki CYCG ve YBMG filtrelerinden elde edilen PSNR ve SSIM değerleri.	51
Çizelge 4.15 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.	52
Çizelge 4.16 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.	53
Çizelge 4.17 : Monarch görüntüsünün farklı yoğunluklardaki farklı filtrelerle elde edilen PSNR ve SSIM değerleri.	54
Çizelge 4.18 : FUMF ve YBMGM9 filtrelerinin farklı yoğunluktaki PSNR değerleri.	57
Çizelge 4.19 : FUMF ve YBMGM9 algoritmalarının farklı gürültü yoğunluklarındaki SSIM değerleri.	58
Çizelge 4.20 : MFUMF ve YBMGMF9 algoritmalarının farklı gürültü yoğunluklarındaki PSNR değerleri.	59

Çizelge 4.21	Farklı gürültü seviyeleri altında Lena görüntüsü için PSNR değerlerinin karşılaştırılması.....	59
Çizelge 4.22	YBMGMF9 filtresinin farklı yoğunluklardaki MS-SSIM değerleri.....	60
Çizelge 4.23	0,2 yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntülerin PSNR ve SSIM değerleri.	60
Çizelge 4.24	0,5 yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntülerin PSNR ve SSIM değerleri.	60
Çizelge 4.25	0,9 yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntülerin PSNR ve SSIM değerleri.	61
Çizelge 4.26	0,9 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin FUMF, YBMGMF9 VE CYCG5 ile elde edilen PSNR ve SSIM değerleri.....	62
Çizelge 4.27	YBMG ile elde edilen algoritmaların 0,1 yoğunluktaki tuz ve biber gürültüsüne sahip görüntüler için PSNR değerleri.	63



ŞEKİL LİSTESİ

Sayfa

Şekil 4.1	: (a) Orijinal görüntü, (b) 0,01 varyanslı Gauss gürültülü görüntü, (c) ÇYÇGMF1 filtresi, (d) ÇYÇGMF2 filtresi, (e) ortalama filtre, (f) medyan filtre	38
Şekil 4.2	: (a) Orijinal görüntü, (b) 0,05 varyanslı alaca gürültülü görüntü, (c) ÇYÇGMF1 filtresi, (d) ÇYÇGMF2 filtresi, (e) ortalama filtre, (f) medyan filtre	39
Şekil 4.3	: (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF1 filtre, (c) YBMGMF2 filtre, (d) medyan filtre.....	40
Şekil 4.4	: (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF1 filtre, (c) YBMGMF2 filtre, (d) medyan filtre.....	40
Şekil 4.5	: (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF3 filtre, (c) YBMGMF4 filtre, (d) medyan filtre.....	41
Şekil 4.6	: (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF3 filtre, (c) YBMGMF4 filtre, (d) medyan filtre.....	41
Şekil 4.7	: (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF5 filtre, (c) medyan filtre.	42
Şekil 4.8	: (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF5 filtre, (c) medyan filtre.....	42
Şekil 4.9	: (a) Orijinal görüntü, (b) 0,1 gürültülü görüntü, (c) YBMGMF6 ile temizlenmiş görüntü.....	44
Şekil 4.10	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF6 ile temizlenmiş görüntü.....	45
Şekil 4.11	: (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	46
Şekil 4.12	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	47
Şekil 4.13	: (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) YBMGMF8 ile temizlenmiş görüntü.....	48
Şekil 4.14	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF8 ile temizlenmiş görüntü.....	48
Şekil 4.15	: (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) ÇYÇGMF3 ile temizlenmiş görüntü, (d) ÇYÇGMF4 ile temizlenmiş görüntü.	49
Şekil 4.16	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) ÇYÇGMF3 ile temizlenmiş görüntü, (d) ÇYÇGMF4 ile temizlenmiş görüntü.	50
Şekil 4.17	: (a) Orijinal görüntü, (b) 0,6 gürültülü görüntü, YBMGMF7 ile temizlenmiş görüntü.....	50
Şekil 4.18	: (a) Orijinal görüntü, (b) 0,7 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	51
Şekil 4.19	: (a) Orijinal görüntü, (b) 0,8 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	52

Şekil 4.20	: (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	52
Şekil 4.21	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	53
Şekil 4.22	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	53
Şekil 4.23	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.....	54
Şekil 4.24	: (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) YBMGMF9 ile temizlenmiş görüntü.....	54
Şekil 4.25	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF9 ile temizlenmiş görüntü.....	55
Şekil 4.26	: (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) YBMGMF9 ile temizlenmiş görüntü.....	55
Şekil 4.27	: (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	56
Şekil 4.28	: (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	56
Şekil 4.29	: (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	57
Şekil 4.30	: Elaine görüntüsünde, sırasıyla (a),(b),(c) ve (d) de 0,2, 0,4, 0,6 ve 0,8 yoğunluklarda tuz ve biber gürültüsü uygulanmış, (e), (f), (g) ve (h) de FUMF kullanılarak, (i), (j), (k) ve (l) de YBMGMF9 filtresi kullanılarak temizlenmiş görüntüler.....	58
Şekil 4.31	: (a) 0,2 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	61
Şekil 4.32	: (a) 0,5 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	62
Şekil 4.33	: (a) 0,9 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	62
Şekil 4.34	: (a) 0,9 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	63
Şekil 4.35	: (a) 0,9 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.....	63
Şekil 4.36	: (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) FUMF ile filtrelenmiş, (d) YBMGMF9 ile filtrelenmiş, (e) ÇYÇGMF5 ile filtrelenmiş	64
Şekil 4.37	: (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) FUMF ile filtrelenmiş, (d) YBMGMF9 ile filtrelenmiş, (e) ÇYÇGMF5 ile filtrelenmiş	64
Şekil 4.38	: FUMF ve YBMGMF9 filtrelerinin farklı yoğunluk değerlerindeki PSNR değerleri.....	65

YÜKSEK BOYUTLU MODEL GÖSTERİLİMİ ve ÇOK DEĞİŞKENLİLİĞİ YÜKSELTİLMİŞ ÇARPIMLAR GÖSTERİLİMİ ile GÖRÜNTÜ ÜZERİNDEKİ GÜRÜLTÜLERİ GİDERME

ÖZET

Bu tez çalışmasında Yüksek Boyutlu Model Gösterilimi (YBMG) ve Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi (ÇYÇG) yöntemleri kullanılarak görüntü üzerindeki gürültüleri giderme çalışması yapılmıştır. Tuz biber gürültüsünü gidermeye yönelik oldukça geniş bir çalışmaya yer verilmekle birlikte gauss ve alacalı gürültülerini giderme konusunda da çalışmalar yapılmıştır.

YBMG ve ÇYÇG yüksek boyutlu fonksiyonların veya verilerin ayrıştırımında kullanılan yöntemlerdir. Karmaşık fonksiyonları veya verileri daha basit ve anlaşılır bileşenlere ayırarak modelleme yapabilme kapasitesine sahiptir ve bilimsel yazında oldukça etkin olarak kullanılmaktadır. Bu yöntemler, yüksek boyutlu veri setlerini daha rahat işlenebilir alt boyutlara indirgeyen birer böl ve yönet algoritması olarak hesaplama kolaylığı sağlamaktadır.

Görüntü üzerinde gürültünün giderilmesi, yapılacak olan işlemin kalitesi ve güvenilirliği açısından büyük önem taşımaktadır. Bu çalışmada görüntü verisinde gürültü olarak isimlendirilen kayıp veya eksik verileri YBMG ve ÇYÇG bileşenlerinden faydalananak en doğru şekilde tahmin etme işlemi amaçlanmıştır. Böylelikle hem literatürün güncel ve iyi çalışmaları kadar etkili yeni yöntemler geliştirilmiş hem de YBMG ve ÇYÇG algoritmaları yeni kullanım alanları bulmuştur.

Çalışmada öncelikle literatürün temel ve güncel filtreleri anlatılmıştır. Gauss gürültüsünü gidermek için kullanılan ortalama filtre, bulanık filtre, gauss görüntü filtresi gibi filtrelerden bahsedilmiş devamında alaca gürültüsü için Lee filtre, Frost filtre, Kuan filtre, doğrusal olmayan uyumlu difüzyon filtre gibi çalışmalara yer verilmiştir. Bu tez çalışmasında üzerinde genişçe durulan tuz ve biber gürültüsü için ise medyan filtre, uyarlanabilir medyan filtre, sürekli kesirler interpolasyon filtresi, piksel yoğunluğu tabanlı filtre, farklı uygulanabilen medyan filtresi, pencere medyan filtre, modifiye edilmiş farklı uygulanabilir medyan filtre gibi literatürün en güncel istatistiksel yöntemlerinin çalışma prensiplerine yer verilmiştir.

Bu tez çalışmasında geliştirilen algoritmada kullanılan YBMG ve ÇYÇG yöntemlerinin, analitik fonksiyonlara, verilere ve görüntülere nasıl uygulandığı anlatılmıştır. YBMG ve ÇYÇG'nin görüntü verisine uygulanarak elde edilen bileşenlerinden faydalananak gürültü giderme algoritmaları oluşturulmuştur. Çalışmada öncelikle alacalı ve gauss gürültülerine değinilmiş bu gürültüler hakkında bilgi verilerek giderilmesi için ÇYÇG tabanlı iki farklı algoritma oluşturulmuştur. Bunlardan ilki

bu gürültüleri gidermek için ÇYÇG'nin sabit teriminden faydalanyorken, ikinci algoritma ÇYÇG'nin tek değişkenli terimlerden faydalanmaktadır.

Tez çalışmasında asıl olarak odaklanılan araştırılan konu tuz ve biber gürültüsünü gidermeye yönelik algoritmaların geliştirilmesidir. Bu amaçla, öncelikle YBMG yönteminden faydalanılmıştır. İlk olarak YBMG açılımının ilk terimleri olan sabit terimden ve birli terimlerden yararlanılması düşünülmüş ve bu terimlerden istatistiksel yöntemlerle elde edilen yeni değer kayan pencere yapısında oluşturulan gürültülü merkez piksel yerine konulmuştur. Çalışmanın sonraki aşamalarında başarının artırılması amacıyla yapıya iteratıflık eklenmiş ve sonuç olarak gerçekten başarısı daha iyi olan yeni bir algoritma elde edilmiştir. Bununla da yetinilmemiş başarının sürekli artabilmesi adına algoritmalar geliştirilmeye devam edilerek toplamda YBMG kullanılarak 9 adet birbirinden farklı ve her biri öncekilere göre daha başarılı olan algoritmalar tasarlanmıştır.

İlk iki yöntem de gürültü tespitinde bütün gürültülü olduğu düşünülen piksel değerlerine 0 atanarak işlem yapılırken, üçüncü yöntemden itibaren tuz ve biber gürültüsünün yapısı gereği 0 veya 255 olan değerler aynen korunarak gürültülü piksel olarak işleme alınmıştır. Yeni oluşturulan gürültü tespiti mekanizması da iteratif olarak uygulanarak gürültü gidermedeki sonuç daha iyi hale getirilmeye çalışılmıştır. İteratif olan yöntemlerden farklı olarak, elde edilen deneyimler yardımıyla, tuz ve biber gürültüsünü yüksek doğruluk oranıyla ve hesaplama maliyetini arttıran iterasyona gereksinim duymayacak şekilde giderebilmek amacıyla yeni bir algoritma daha geliştirilmiştir.

Geliştirilen algoritmaların tümü düşük pencere boyutları ile çalışarak tamamlanmış ancak çalışmanın ilerleyen safhalarında pencere boyutlarında yapılan değişikliklerin performansı doğrudan etkilediği gözlemlenerek bu konuya yoğunlaşılmıştır. Özellikle yüksek yoğunluklu gürültülerin içerildiği gürültülerde pencere boyutunda yapılan değişikliklerin başarıyı arttırdığı görülmüş ve bu amaçla ele alınan her bir pencere simetrisi yardımıyla büyütülmüş ve bu büyütülen pencereler üzerinde YBMG bileşenleri bulunmuş ve bu bileşenler kullanılarak elde edilen değerler gürültülü pikseller yerine konularak tuz ve biber gürültüsü giderilmeye çalışılmıştır. Pencere büyütme işlemi düşük yoğunluklarda çok başarılı sonuçlar vermekle birlikte yüksek yoğunlukta literatürdeki çalışmaları geçecek kadar başarılı olamadığından yeni bir arayış içine girilmiş ve sonunda dinamik pencere yapısına geçilerek başarının artırılması hedeflenmiştir. Dinamik pencereler kullanımı sırasında düşük boyutta orijinal piksel değeri bulunamadığı takdirde pencere büyütülmüştür. Orijinal piksel değeri elde edilene kadar bu genişletme işlemine devam edilmiş ve böylelikle yüksek yoğunluk düzeylerinde de başarılı çalışır hale gelmiştir.

Çalışmanın son kısmında YBMG kullanılarak yapılandırılan algoritmaların başarısının ÇYÇG ile de yakalanıp yakalanmayacağı merak konusu olduğundan algoritmalar ÇYÇG kullanılarak yeniden yapılandırılmıştır. ÇYÇG ile yapılan ilk çalışmada pencere boyutu büyütülerek elde edilen YBMG algoritmasının benzeri uygulanmıştır. Devamında ÇYÇG ve YBMG yöntemlerinin bir arada kullanıldığı durumda daha iyi sonuçlar elde edilebileceği düşünülerek YBMG yardımı ile bir kere filtrelenmiş görüntüden elde edilen destek fonksiyonları ile görüntü ikinci kez ÇYÇG ile filtrelenmiştir fakat bu durum sonuçlara çok büyük bir katkı sağlamamıştır. ÇYÇG ile elde edilen son yöntemde ise dinamik pencere yapısı kullanılarak sonuçlar iyileştirilmiştir. ÇYÇG ile elde edilen algoritmalar YBMG ile oluşturulan

algoritmalarla çok yakın sonuçlar vermekle birlikte gri ton görüntülerde YBMG'nin daha başarılı olduđu söylenebilir.

YBMG ile en son geliştirilen algoritma farklı uygulanabilir medyan filtre (FUMF), piksel yoğunluđu tabanlı filtre (PYTF), uyarlanabilir türev tabanlı genelleştirilmiş kesirli algoritma (UTTKA) ile kıyaslandığında daha iyi sonuçlar vermiş, YBMG'yi takip eden en iyi sonuçları da ÇYÇG ile elde edilen son algoritma vermiştir.

Renkli görüntüleme yöntemlerinin uygulanmasında her bir görüntü kanalının (RGB) ayrı ayrı işlenmesi gerektiği göz önüne alınmıştır. Yüksek Boyutlu Model Gösterilimi (YBMG) ve Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi (ÇYÇG) yöntemleri, bu kanalları bağımsız olarak ele alarak her kanal için ayrı ayrı bileşenler çıkarmış ve gürültü giderme işlemlerini gerçekleştirmiştir. Bu şekilde, her bir kanalın bağımsız olarak işlenmesi, yöntemlerin renkli görüntülerde de düşük ve yüksek yoğunluklardaki başarılarını göstermiş ve tezin kapsamında elde edilen sonuçların geniş bir uygulama yelpazesine hitap ettiğini ortaya koymuştur. Renkli görüntüler üzerinde yapılan çalışmalar literatürde çok geniş yer almadığı için sonuçlar görsel olarak verilmiştir.





IMAGE DENOISING via HIGH DIMENSIONAL MODEL REPRESENTATION and ENHANCED MULTIVARIATE PRODUCT REPRESENTATION

SUMMARY

In this thesis study, noise removal on the image was carried out by using High Dimensional Model Representation (HDMR) and Enhanced Multivariate Product Representation (EMPR). First of all, the study of removing gaussian and speckle noise with EMPR was briefly included, and then a very extensive study was carried out to eliminate salt and pepper noise with both EMPR and HDMR.

Noise reduction in images is of great importance for the quality and reliability of the process to be performed. In this study, the aim is to accurately predict the missing or lost data in image data, referred to as noise, by utilizing components of HDMR and EMPR. In this way, new methods as effective as the current and well-established studies in the literature have been developed, and HDMR and EMPR algorithms have found new areas of application. The filters to be produced in this study are expected to contribute to various fields where image processing is used.

The study first includes the current filters in the literature. Filters used to remove Gaussian noise, such as the mean filter, fuzzy filter, and Gaussian image filter, are described. Subsequently, works on filters for speckle noise, such as the Lee filter, Frost filter, Kuan filter, and nonlinear coherent diffusion filter, are included. For salt-and-pepper noise, which is extensively covered in this thesis, the most current statistical methods in the literature are discussed, including the median filter, adaptive median filter, continued fractional interpolation filter, based on pixel density-based filter, different applied median filters, window median filter, and modified different applied median filters.

The HDMR algorithm used in the study is first described in terms of how it is applied to analytical functions. Subsequently, its application to functions in Euclidean space, where the values are known at a discrete finite number of points, i.e., to data, is explained. Finally, considering the image as a three-dimensional data set, how the algorithm is applied to images is described. Similarly, the application of the EMPR algorithm to functions and data is also explained.

In developing the noise reduction algorithms, the components obtained by applying HDMR and EMPR to image data were utilized.

Firstly, using EMPR, the EMPR1 and EMPR2 algorithms were created to reduce speckle and Gaussian noise in images. The EMPR1 algorithm leverages the constant term of EMPR to mitigate these noises, while the EMPR2 algorithm uses univariate terms. The operation of these algorithms involves dividing the image data into 3×3

square windows and predicting the value of the central pixel using the components obtained from EMPR.

The study extensively focuses on reducing salt-and-pepper noise using an algorithm called HDMR1, which utilizes the univariate terms and constant term of HDMR. All values considered noise, 0 or 255, are initially converted to 0. Then, for the points identified as 0 in the image, a 3×3 window is created with the 0 value at the center. The central pixel value is predicted using the constant and univariate terms of HDMR. The subsequent filter, HDMR2, involves the iterative application of the HDMR1 filter. This approach aims to identify noisy pixel values that could not be detected in a single iteration in the subsequent iterations. In the HDMR3 filter, pixel values of 255 are not converted to 0 before processing. 3×3 windows are created around the pixels identified as 255 or 0, and the central pixel is predicted. HDMR4 filter is an iterative application of HDMR3 filter. Unlike iterative methods, the HDMR5 algorithm was created to detect all noisy points at once. Here, while browsing through the pixels, a preliminary filtering is performed on all pixels that are considered to be noisy. Then, all of them are estimated at the same time based on the HDMR components. In order to improve the studies carried out with HDMR, a hybrid model was created to obtain more accurate results in places where the transition in pixel values is sudden and rapid. This new algorithm is named HDMR6 in this study. In the HDMR6 algorithm, the values 0 or 255, which are considered to be noisy, are accepted as the center pixel and the surrounding area is taken as a 3×3 window. By calculating the standard deviation of the window, the value obtained with HDMR constant and unity terms was used where the standard deviation was less than the threshold value obtained experimentally, and a non-repetitive median filter was used where the standard deviation was greater than the threshold value. The image was improved by applying the algorithm iteratively. The studies carried out so far have been successful in removing low intensity salt and pepper noise. The studies carried out so far have been successful in removing low intensity salt and pepper noise. In order to be successful when the intensity increases, HDMR7 algorithm has increased the window size to 9×9 by utilising the symmetries of 3×3 windows. This expansion was achieved by copying the 3×3 window to the right, left, up, down and diagonally. When the HDMR algorithm was applied to this 9×9 window, it was observed that the result improved. By combining the methods given in HDMR6 and HDMR7, a new method called HDMR8 is obtained. In this method, firstly, the 3×3 size windows, which accept the noisy value as the centre, are enlarged to 9×9 size with the help of their symmetries, and then the mean and standard deviation of the noiseless pixels are calculated. After replacing the noisy pixel values with the mean, if the standard deviation is above the optimal threshold value, the centre pixel value in the window is calculated by applying a non-repetitive median to the window, if not, the noise removal algorithm created with HDMR is applied.

In the continuation of the study, the efficiency of the algorithm was increased at high densities. In order to obtain good results at high density, the window size in the filter has been made adaptive. Thus, the original pixel value, which cannot be found when the window size is small, can be found when the window size is large. For this purpose, firstly, the image size was increased by copying the symmetries of pixel values of three units each to the right, left, up, down and corners in the whole image. Thus, if the first pixel is noisy, it is first checked whether there is an original value in a 3×3 window around it, if not, the window size is increased to 5×5 , and if there is no noiseless pixel here, the size is increased to 7×7 . Then the window was shifted and the same

algorithm was applied to the whole image. The expansion process of three units each made the entire image from the first pixel to the last pixel suitable for the 7×7 window size. Thus, even at very high intensity, the original pixel value was captured and the algorithm gave very good results. After the algorithm was applied once, the cleaned image was run three more times for a 3×3 window by taking one-pixel symmetries of the cleaned image in case there were any noisy pixels that could not be captured. In this way, all noisy pixel values were captured and removed even at very high intensity values. Highly successful results were obtained from the HDMR9 algorithm and it was applied on both grey scale and colour images.

The EMPR3 algorithm, EMPR4 algorithm and EMPR5 algorithm were inspired by the algorithms created with HDMR. These are the adaptations of HDMR7, HDMR8 and HDMR9 filters to EMPR respectively. The algorithms obtained with EMPR are also applied to both grey tone and colour images and the experimental results obtained are given in the related section.

The results obtained from the filters generated with HDMR and EMPR are presented with three different metrics. The metrics used for quantitative analysis are Peak Signal-to- Noise Ratio (PSNR) and Structural Similarity (SSIM). PSNR and SSIM values are metrics that show how close the resulting image is to the original image. A higher PSNR value indicates that the resulting image is closer to the original image. SSIM, on the other hand, takes values between 0 and 1 and as it increases, the quality of the image improves. The MS-SSIM metric was also used to show the achievements in edge detail. In the MS-SSIM value, the success increases as the value approaches 1.

According to the results obtained, EMPR2 algorithm works better than the mean and median filter for Gaussian and speckle noise. For gray scale images with 0.1 salt and pepper noise, the HDMR2 algorithm works better than the median filter. When HDMR3 and HDMR4 filters are compared among themselves, HDMR4 filter gives better results. HDMR5 filter is inferior to HDMR4 filter but better than median filter. For images with 0.1 salt and pepper noise, HDMR1, HDMR2, HDMR3, HDMR4 and HDMR5 are compared with each other and the best result belongs to HDMR4 algorithm. The hybrid model HDMR6 filter did not outperform the DAMF when compared with important studies in the literature. The HDMR7 filter was able to pass the DAMF at an intensity level of 0.2. Similarly, the HDMR8 filter surpassed DAMF in both PSNR and SSIM scales. However, HDMR7 and HDMR8 algorithms did not exceed all other algorithms after 0.5 density level. The EMPR3 and EMPR4 filters did not perform as well as the HDMR7 and HDMR8 filters at 0.2 density. However, EMPR3 and EMPR4 also gave good results as the density increased. The final version of the study, HDMR9 algorithm outperformed all the compared filters at all intensity levels. The EMPR5 algorithm also outperformed the other studies at 0.2 intensity, but falls behind YBMG9 for grey level images. However, as the intensity level increased, there were some images where the EMPR5 algorithm gave better results than the YBMG9 algorithm. When the denoised images are analysed, both HDMR9 and EMPR5 algorithms successfully removed the noise at the all intensity levels. Since there is no extensive source in the literature for color images, the study was not compared with different filters. However, when looking at the images obtained, it is observed that the color images are very well cleaned of noise. It is seen that the HDMR algorithm is especially successful in preserving edge details in both grayscale and color images.



1. GİRİŞ

Görüntü üzerindeki gürültü, görüntünün oluşumu, depolanması veya taşınması esnasında oluşabilecek görüntü kalitesinin bozulmasına sebep olan eksiklikler veya kayıplar olarak tanımlanabilir. Görüntü üzerinde oluşan bu gürültüleri giderme işlemi ise oluşan veri kayıplarını en doğru şekilde tahmin etme işlemidir [1].

Gürültüler, görüntünün netliğini ve kalitesini düşürebilir. Bu, fotoğraflarda veya videolarda istenmeyen pikseller veya bulanıklık olarak ortaya çıkabilir. Gürültülerin azaltılması, daha net ve daha estetik olarak hoş görünen görüntüler elde etmeye yardımcı olur. Gürültüler, görüntüdeki önemli bilgileri maskeleyebilir veya bozabilir. Özellikle tıbbi görüntüler veya bilimsel görüntülerde, gürültüler gerçek verileri gizleyebilir veya yanıltıcı sonuçlara yol açabilir. Bu nedenle, gürültülerin azaltılması, doğru bir analiz ve yorumlama için gereklidir. Otomatik görüntü işleme sistemleri, gürültülerin varlığından etkilenebilir. Örneğin, nesne tanıma veya yüz tanıma gibi uygulamalarda, gürültüler yanlış algılamalara veya tanımaya neden olabilir. Gürültüyü azaltmak, bu tür sistemlerin doğruluğunu ve güvenilirliğini artırabilir. Gürültüler, görüntülerin sıkıştırılması veya depolanması sırasında gereksiz veri miktarına yol açabilir. Gürültüyü azaltmak, veri boyutunu küçülterek depolama alanından tasarruf sağlayabilir veya iletim sırasında bant genişliğini azaltabilir. Bu nedenlerle, görüntü üzerindeki gürültülerin giderilmesi, birçok uygulama ve endüstride önemli bir adımdır.

Görüntü üzerinde en sık rastlanan gürültüler tuz ve biber gürültüsü, Gauss gürültüsü, speckle gürültüsü gibi gürültülerdir. Tuz ve biber gürültüsünde piksel değerleri maksimum veya minimum değer olarak görüntüde bozulmalara yol açar. Siyah beyaz bir görüntüde gri seviye değerleri 0 ile 255 arasında değişir. Piksel değeri 0 ya da 255 olduğunda görüntü üzerinde siyah veya beyaz noktacık şeklinde tuz ve biber gürültüsü oluşturur. Tuz ve biber gürültüsüne veri iletimindeki hatalar sebebiyle oluşan ani değişiklikler neden olur [2]. Çarpımsal bir gürültü olan alacalı gürültü daha çok radar, ultrason, lazer, sonar gibi görüntülerde oluşur. Tanecikli bir yapıya sahiptir ve görüntüde bozulmalara sebep olur [3]. Gauss gürültüsü normal dağılıma

sahip olan bir gürültü türü olup bir çok doğal ve yapay süreçte ortaya çıkabilir [4]. Elektronik cihazlarda veya sensörlerdeki ısı hareketinden kaynaklanan termal gürültü, genellikle Gauss dağılımını izler. Görüntü işleme sürecinde gerçekleşen matematiksel işlemlerdeki yuvarlama hataları veya dijital veri işlemenin diğer yan etkileri, Gauss gürültüsüne benzer bir dağılım gösterebilir. Optik sistemlerdeki yansımalar veya ışığın farklı yüzeylerden yayılması sırasında görüntüye eklenen gürültü, genellikle Gauss gürültüsüyle karakterizedir.

Bu çalışmada geniş olarak tuz ve biber gürültüsünü gri ton ve renkli görüntülerden gidermeye yönelik yapılan çalışmaların yanında gauss ve alaca gürültülerini gri ton görüntülerden giderilmesine yönelik çalışmalara da yer verilmiştir.

1.1 Tezin Amacı ve Kapsamı

Çalışmamızda, Yüksek Boyutlu Model Gösterilimi (YBMG, High Dimensional Model Representation, HDMR) ve Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi (ÇYÇG, Enhanced Multivariate Product Representation, EMPR) kullanılarak görüntü üzerinde oluşan farklı gürültü çeşitlerinin giderilmesi amaçlanmıştır. Üzerinde çalışılan gürültüler genişçe tuz ve biber olmakla beraber bunun yanında gauss gürültüsü ve alacalı gürültüdür. Bu çalışmanın amacı görüntüden gürültüleri mevcut diğer yöntemler kadar iyi arındırabilen filtreler geliştirmektir. Geliştirilen bu filtreler görüntüden gürültüyü arındırırken, gürültülü piksel noktalarını yüksek doğrulukla tahmin etmeyi amaçlamakta ve görüntü içindeki doğru veriye sahip gürültüsüz piksel değerlerini saklayarak görüntü detaylarını korumayı hedeflemektedir.

1.2 Özgün Değer ve Yaygın Etki

Yüksek Boyutlu Model Gösterilimi ve Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi farklı farklı alanlarda kullanılabilen birer böl-ve-yönet algoritmasıdır. Bu iki yöntem görüntü filtrelemede daha önce kullanılmamıştır. Özgün olarak elde edilecek yeni filtrelerle YBMG ve ÇYÇG algoritmaları yeni bir alanda kullanılacaktır. Görüntü üzerinden farklı gürültü çeşitlerini gidermek için de yeni algoritmalar elde edilmiş olacaktır. Çalışmada üretilecek olan filtreleme algoritmalarının görüntü

işlemenin kullanıldığı alanlarda oluşan gürültü kaynaklı problemlerin çözülmesine katkı sağlaması beklenmektedir.

1.3 Literatür Özeti

Görüntü üzerindeki gürültüleri gidermeye yönelik yapılan çalışmalar, literatürdeki güncel haliyle ele alınmıştır. Çalışmada gauss, alacalı ve tuz ve biber gürültülerine yer verildiği için bu gürültüler için oluşturulan yöntemler özetlenmiştir.

Ortalama filtre gauss gürültüsünü gidermek için kullanılan doğrusal bir filtredir [5]. Ortalama filtrede seçilen boyuttaki pencere içindeki piksel değerlerinin ortalaması hesaplanır ve pencerenin ortasındaki piksel değeri olarak atanır [6]. Doğrusal filtreler keskin kenarları bulanıklaştırma, çizgileri yok etme eğilimindedirler. Dolayısıyla ortalama filtre kenar detaylarını korumakta başarılı bir filtre değildir. Bunun üstesinden gelebilmek için doğrusal olmayan filtreler görüntü detaylarını koruyabildiği için daha çok kullanılmaya başlanmıştır [7]. Gauss gürültüsünü gidermek için kullanılan bulanık filtre (fuzzy filter) doğrusal olmayan bir filtredir [5]. Bulanık filtrenin amacı yüksek kontrastlı bir görüntü elde etmektir.

Görüntüden gauss gürültüsünü arındırmak için iki aşamadan oluşan bulanık filtre geliştirilmiştir [8]. Birinci aşamada çalışılan piksel etrafında 8 yönde bulanık bir gradyan hesaplar. İkinci aşamada komşu piksel değerlerini ve bu gradyanı kullanarak yumuşatma gerçekleştirir.

Gauss gürültüsünü gidermek için gauss fonksiyonuna bağlı fakat diğer parametreleri kullanıcı tarafından belirlenen gradyan uyarlamalı gauss görüntü filtresi tasarlanmıştır [1]. Bu filtre görüntüdeki her bir piksel değeri için tek bir varyans değeri yerine ayrı ayrı varyans değeri hesaplar ve farklı çekirdek oluşturur. Gradyan uyarlamalı gauss gürültüsü ortalama filtre ve gauss filtresinden daha başarılıdır [9].

Alacalı gürültü daha çok radar, ultrason, laser gibi görüntülerde ortaya çıkan çarpımsal bir gürültüdür [3]. Son yıllarda alacalı gürültüyü gidermek için bir çok yöntem geliştirilmiştir. Bu yöntemlerden bazıları Lee [10], Kuan [11], Frost [12], değiştirilmiş Lee filtre [13], değiştirilmiş Frost filtre [13] ve anizotropik difüzyon filtresidir [14].

Lee'nin filtresi [10] yerel gradyan bilgisini kullanır. Görüntüyü modellemeyi gerektirmez, kenarları ve ince ayrıntıları korur. Yerel ortalama ve varyans kenarın

yönelimine bağı olarak hesaplanır. Böylelikle kenar boyunca gürültü giderilir ve kenar keskinlikleri korunmuş olur.

Kuan'ın filtresi [11] durağan olmayan ortalama ve durağan olmayan varyans üzerine kurulan bir modellemedir. Durağan olmayan ortalama, bir görüntünün bütün yapısını tanımlar ve durağan olmayan varyans, görüntünün kenar ve temel doku bilgisini karakterize eder. Genellikle durağan olmayan istatistik verileri önceden mevcut değildir, görüntünün bozulmuş olması sonucu ortaya çıkar. Uyarlanabilir görüntü yumuşatma filtresi durağan olmayan görüntü modelleme yaklaşımı ile bozulmuş görüntünün yerel istatistikleri hesaplayarak görüntüyü yeniden oluşturur.

Frost'un filtre modeli [12] çarpımsal gürültülü radar görüntüleri üzerine yapılmış bir çalışmadır. Çalışma yeni bir model ve model bazlı bir görüntü iyileştirme tekniği sunar. Uyumlu görüntüleme tekniklerini kullanan mikrodalga sensörleri için tasarlanmış bir tekniktir. Kullanılan yöntem görüntü verilerini yeniden tahmin etmek için minimum ortalama kare hata MMSE (minimum mean square error) kullanır. MMSE filtresinin tasarımı, küresel olarak durağan olmayan görüntünün durağan olduğu varsayımına dayanır.

Doğrusal olmayan uyumlu difüzyon (nonlinear coherent diffusion NCD) modeline dayalı olarak ultrason görüntülerinde gürültü azaltmaya yönelik bir yaklaşım da bulunmaktadır [14]. Burada önerilen model üç farklı modeli birleştirmektedir. Gürültü boyutuna ve görüntü anizotropisine göre model, izotropik difüzyondan anizotropik uyumlu difüzyona geçen model ve son olarak ortalama eğrilik hareketine doğru aşamalı olarak değişir. Bu tekniğin teorisi hayali ve klinik ultrason görüntülerine uygulanmış ve önceki tekniklerle sonuçlar kıyaslandığında başarılı sonuçlar vermiştir.

Tuz ve biber gürültüsünün giderilmesi için literatürde en çok bilinen filtre medyan filtredir [15]. Medyan filtre gibi bir çok doğrusal olmayan filtre tuz ve biber gürültüsünü temizlemek için kullanılmaktadır. Literatürde medyan filtrenin yanısıra tuz ve biber gürültüsünü temizlemek için geliştirilen pek çok yöntem vardır. Bu yöntemlerden bazıları uyarlanabilir medyan filtre, UMF (Adaptive median filter AMF) [16], uyarlanabilir ağırlıklı ortalama filtre, UAOF (Adaptive weighted mean filter AWMF) [17], sürekli kesirler interpolasyon filtresi, SKIF (Continued fractions interpolation filter CFIF) [18], küresel uyarlanabilir integral algoritması KUIA (Global

Adaptive Fractional Integral Algorithm, GAFIA [19], piksel yoğunluğu tabanlı filtre, PYTF (Based on pixel density filter, BPDF) [20], farklı uygulanabilen medyan filtresi, FUMF (Different Applied Median Filter, DAMF) [2], uyarlanabilir türev tabanlı genelleştirilmiş kesirli gürültü giderme algoritması, UTTKA (Generalized Fractional Derivative based Adaptive Algorithm for Image Denoising, GFDBA) [21] gibi sonradan üretilen ve medyan filtreye göre daha iyi sonuçlar verebilen filtrelerdir. Ayrıca Dört Aşamalı Medyan-Ortalama Filtre (Four stage median-average filter) [22], Modifiye Edilmiş Kademeli Filtre, MEKF (Modifed Cascaded Filter, MCF) [23], iki aşamalı filtre (two-stage filter) [24] gibi filtreler de tuz ve biber gürültüsünü gidermekte etkilidir.

Medyan filtre çalışılan penceredeki piksel değerlerinin ortancasını ortadaki piksel değeri yerine koyarak çalışır [15]. Medyan filtrede kayan pencereler bütün piksel değerleri üzerinde gezer ve bütün piksellerde değişime yol açar. Bu durumda gürültüleri arındırırken aynı zamanda görüntünün genelinde bir bozulmaya sebep olur. Medyan filtre görüntünün kenar özelliklerini kaybettirmeden gürültüleri temizlediği için önemli bir yöntemdir. Fakat ilgili piksel değerinin gürültülü olup olmadığına bakmaksızın bütün piksel değerleri üzerine uygulanır. Bu durum da görüntüde genel bir bozulmaya sebep olmaktadır. Ayrıca medyan filtre düşük yoğunluklu gürültülerde başarılı sonuçlar verirken, gürültü yoğunluğunun artmasıyla görüntüdeki temizlenme önemli ölçüde kötüleşmektedir [25, 26].

Medyan filtreyi iyileştirmek için uyarlamalı medyan filtre (adaptive median) geliştirilmiştir [16]. Uyarlanabilir medyan filtre yöntemi iki aşamalıdır. Birinci aşama pikselleri gürültülü ve gürültüsüz olarak sınıflandırmayı amaçlar. İkinci aşamada ise gürültüsüz olduğuna karar verilen pikseller hiçbir değişikliğe uğramadan aynen çıktı görüntüsüne kopyalanır. Medyandan farklı olarak uyarlamalı pencereler kullanılır. Gürültüyü gidermek için kayan pencerelerde medyan değeri bulunurken yalnızca gürültüsüz olduğuna karar verilen pikseller dikkate alınır. Hem düşük hem de yüksek gürültü yoğunluğunda iyi sonuçlar verebilmekle beraber bazen gürültülü pikselleri kaçırabilmektedir. Bu durumun üstesinden gelebilmek için uyarlanabilir ağırlıklı ortalama filtre (Adaptive Weighted Mean Filter AWMF) geliştirilmiştir [17]. Uyarlanabilir ağırlıklı ortalama filtre uyarlanabilir pencerelerde gürültüsüz olduğu düşünülen piksellerin ortalamalarını merkezdeki piksel değeri yerine koyarak çalışır.

Bu ortalama alınırken gürültüsüz piksellerin merkez piksele olan ağırlıkları hesaba katılarak ağırlıklı bir ortalama bulunur.

Tuz ve biber gürültüsünü gidermek için geliştirilen yöntemlerden bir tanesi de sürekli kesirler interpolasyon filtresidir (Continued Fractions Interpolation Filter CFIF) [18]. Sürekli kesirler interpolasyon filtresi pencere gerektirmeyen bir yöntemdir. Gürültülü piksel değerini düzeltmek için tek bir medyan değeri yerine seçilen tüm piksel değerlerini birlikte kullanır. Algoritma yatay, düşey ve çapraz yönlerdeki ağırlıkları ve kesirli interpolasyon kullanarak piksel değerini yeniden hesaplar.

Küresel uyarlanabilir kesirli integral algoritması, KUIA (Global Adaptive Fractional Integral Algorithm, GAFIA) bir başka gürültü giderme yöntemidir [19]. Uyarlanabilir kesirli integral algoritması görüntünün her pikseline kesirli integral uygular ve sekiz yönde piksellerin ortalama gradyanına göre en uygun sıralamayı bulur. Ardından pikseller uygun sıralamadaki eğrilerin frekansına göre işlenir. Bu filtre tuz ve biber gürültüsünü gidermekte iyi sonuçlar vermektedir.

Tuz ve biber gürültüsünü gidermede piksel yoğunluğuna bağlı bir yöntem de geliştirilmiştir (Based on Pixel Density Filter BPDF) [20]. Bu yöntemin de ilk adımı gürültülü olan ve olmayan pikseli belirlemektir. Daha sonra gürültülü piksel merkezde kalacak şekilde uyarlamalı bir pencere oluşturulur. En çok tekrar eden gürültüsüz piksel değeri yeni piksel değeri olarak atanır. Düşük ve orta yoğunlukta bir çok yöntemden daha iyi sonuçlar vermektedir.

2020 yılında yapılan bir çalışma olan uyarlanabilir türev tabanlı genelleştirilmiş kesirli gürültü giderme algoritmasıdır (Generalized Fractional Derivative based Adaptive Algorithm for Image Denoising) [21]. Bu algoritma kesirli filtreler dayalı bir algoritmadır. Kesirli filtreler geliştirilen kesirli operatör ile türetilir. Gürültülü pikseller piksel yoğunluğuna göre tanımlanır ve bunlar uyarlamalı kesirli integral maskeleri ile yeniden hesaplanır. Gürültüsüz pikseller ise uyarlamalı kesirli diferansiyel maske ile işlenir. Görüntünün varyansı ve gradyan özellikleri ile kesirli maskeler belirlenir. Performans olarak diğer birçok yöntemden daha iyi sonuçlar vermektedir.

Son dönem çalışmaları içerisinde Yüksek Yoğunluklu Tuz ve Biber Gürültüsü için Dört Aşamalı Medyan-Ortalama Filtre (Four stage median-average filter for healing high density salt and pepper noise corrupted images) vardır. Bu algoritma çok

yüksek yoğunlukta da çok iyi sonuçlar vermektedir. Bu algoritma öncelikle iki paralel kısaltılmış medyan filtre (TMF) kullanır. Devamında gürültülü piksele göre maskeleme yapar. Bulanıklaşma etkisini azaltmak için küçük pencere boyutları ile TMF uygular. İkinci aşamada yüksek maske boyutlarına TMF uygular. Kalan gürültülü pikseller üçüncü aşamada çalışan ortalama filtresi ile iyileştirilir. Son aşamada ise sınırdaki gürültülü pikseller iyileştirilir. [22].

Yüksek yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntüler için yapılan çalışmalardan biri Modifiye Edilmiş Kademeli Filtre'dir (Modifed Cascaded Filter MCF). MCF kırılmış medyan değeri kullanarak gürültülü piksel değerlerini iyileştirir. Seçilen pencerede DMF (decision based median filter) kullanılarak gürültülü pikseller bulunur. Gürültülü piksel değerleri, simetrik olmayan kırılmış ortalama filtre kullanılarak seçilen penceredeki değerlerin ortalaması ile değiştirilir. Standart medyan filtreden ve DMF den daha iyi PSNR değerine sahiptir [23].

Siyah beyaz görüntüler için yapılan son dönem çalışmalarından biri ise Yüksek Yoğunluklu Tuz ve Biber Gürültüsü için İki Aşamalı Filtre'dir (A two-stage filter for high density salt and pepper denoising). Hem düşük yoğunlukta, hem yüksek yoğunlukta oldukça başarılı sonuçlar veren bir filtredir. Birinci aşamada orijinal piksellerin medyanını kullanarak yüksek yoğunluklu gürültüyü azaltır. İkinci aşamada düşük yoğunluklu gürültüyü kaldırmak için en çok tekrar eden piksel değerlerinin medyanını kullanır [24].

Renkli görüntülerdeki yüksek yoğunluklu tuz ve biber gürültüsünü gidermek için bir algoritma geliştirilmiştir. Bu algoritmada, RGB renk modelindeki her bir kanal (kırmızı, yeşil ve mavi) ayrı ayrı iki boyutlu matrisler olarak ele alınmış ve her kanal 3x3 boyutlarında pencerelere bölünerek işlenmiştir. Pencere içerisindeki gürültülü piksel noktaları, çevredeki orijinal piksel değerlerinin ortalamasıyla değiştirilerek temizlenmiştir. Eğer pencere içindeki bütün piksel değerleri gürültülü ise, bu durumda bir önceki pencerede elde edilen ortalama değer kullanılmıştır. Hem düşük hem yüksek yoğunlukta oldukça iyi PSNR değerleri elde edilmiştir [27].

Pencere medyan filtre (Window Median Filter WMF) renkli görüntülerden tuz ve biber gürültüsünü arındırmak için yapılan yeni çalışmalardan bir diğeridir. RGB bantlarının her biri için, pencerelerde gezerken gürültülü olan pikselleri 0, gürültüsüz

olan pikselleri 1 değeri ile değiştirir. Gürültülü veriyi içeren matris ile, 0 veya 1 değerlerine dönüştürülmüş matrisi kıyaslayarak, 1 olan değerlere yeni piksele geçer, 0 olan değerlerde pencere oluşturur. Gürültülü piksel değeri dışındaki piksellerin medyanını hesaplar [28].

Lokal olmayan uyarlamalı ortalama filtre (A Nonlocal Adaptive Mean Filter for Removal of Salt-and-Pepper Noise NAMF) siyah beyaz görüntülerdeki tuz ve biber gürültüsü gidermek için geliştirilmiştir. Seçilen pencerede global minimum ve global maksimum değerleri muhtemel gürültülü piksel olarak düşünülür. Uygun bir eşik değeri belirlenerek bu eşik değerine göre gürültülü veya değil diye pikseller belirlenir. Gürültülü olduğu düşünülen piksel, komşu piksel değerlerinin ortalaması ile değiştirilir [29].

Tuz ve biber gürültüsü için üstel ağırlıklı ortalama filtre (Exponentially Weighted Mean Filter for Salt-and-Pepper Noise Removal EWMF) en son yapılan çalışmalardan biridir [30]. Bu filtre mesafeye bağlı, üstel olarak bir ortalama belirler. Gürültülü pikselleri algılayarak ve olası gürültü yoğunluğuna dayalı olarak bir eşik değeri belirler. Düşük yoğunluk dereceleri için Von Neumann komşuluğundaki üstel ağırlıklı ortalamayı, orta veya yüksek yoğunluk dereceleri için Moore komşuluğundaki üstel ağırlıklı ortalamayı kullanır.

Görüntülerdeki yüksek yoğunluklu tuz ve biber gürültüsünün giderilmesi için Uyarlanabilir Yakınlık Yaklaşımlı Piksel Değiştirme algoritması (An Adaptive Content based Closer Proximity Pixel Replacement algorithm) 2020 yılında yapılmış yeni bir çalışmadır [31]. Algoritma, işlenen piksel ile pencere içindeki gürültülü olmayan piksel sayısı arasındaki Öklid mesafesini bulur. Pencere içinde en az iki tane gürültüsüz piksel bulunması yeterlidir. Öklid mesafesine dayalı olarak belirlenen penceredeki gürültüsüz piksellerin medyanı ile gürültülü piksel değerleri belirlenir. Gürültüsüz piksel değeri bulamadığı durumda pencere boyutunu iki katına çıkarır.

Yüksek yoğunlukta iyi sonuç veren yöntemlerden biri de İteratif Ağırlıklı ortalama filtresidir, IAOF (Iterative Weighted-Mean Filter, IWMF) [32]. İlk olarak görüntüdeki gürültülü pikselleri belirler. İkinci aşamada uyarlamalı pencerelerle ağırlıklı ortalama kullanarak gürültülü piksel değerlerini bulur. Devamında bu yöntem yinelemeli olarak tekrar eder.

Kenarları ve ince detayları koruyarak tuz ve biber gürültüsünü gidermek için iki aşamalı olarak geliştirilen yöntem Piksel Tabanlı Segmentasyon ve Uyarlanabilir Medyan Filtresi Kullanarak Tuz ve Biber Gürültüsünü Giderme (Salt and Pepper Noise Removal using Pixel-based Segmentation and Adaptive Median Filter) algoritmasıdır [33]. İlk aşamada muhtemel gürültülü pikseller piksel tabanlı segmentasyon kullanarak tahmin edilir. İkinci aşamada bu pikseller uyarlanabilir medyan filtre ile düzeltilir.

Tuz ve biber gürültüsü için eşikleme ve düzenleme kullanarak geliştirilen teknik, toplam varyasyon düzenlemesine ve tuz ve biber gürültüsünün özelliklerine dayalı olarak gürültüleri kaldırır [34]. Gürültülü pikselleri tespit etmek için gürültü özelliklerinden faydalanır. Bulunan gürültülü pikselleri bulmak için değiştirilmiş bir toplam varyasyon düzenleme tabanlı bir yöntem uygular.

Tuz ve biber gürültüsünü gidermek için en başarılı filtrelerden biri Farklı Uygulanan Medyan Filtre, FUMF'dir (Different Applied Median Filter, DAMF) [2]. Bu filtre uyarlamalı pencereler kullanarak aranan piksel değeri için gürültüsüz değer kullanmayı garanti eder. Bunu yapmaktaki amaç yüksek yoğunlukta küçük boyutlu pencerelerde orijinal değer bulunmama olasılığıdır. FUMF literatürün en iyi sonuç veren çalışmalarından biridir. Tuz ve biber gürültüsünü gidermek için yapılan en yeni çalışmalardan biri Modifiye Edilmiş Farklı Uygulanan Medyan Filtre, MFUMF (A modified form of different applied median filter for removal of salt and pepper noise, MDAMF) 2023 yılında yayımlanmıştır [35]. İki aşamalı bir yöntem olan bu filtre önce piksel değerinin gürültülü olup olmadığını belirler, devamında 7x7 boyutuna genişleyebilen pencereler kullanarak gürültüyü kaldırır. FUMF'ye çok benzeyen bir algoritmadır ve sonuçları FUMF'ye çok yakındır. Pencere boyutuna içerisinde gürültüsüz piksel olup olmadığına bakarak karar verir. Kalan piksellerin medyanını alarak çalışır.

Karar Tabanlı Algoritma KTA (Decision based algorithm DBA), medyanla çalışan bir filtredir. Medyan değerleri gürültülü ise komşu piksellerin ortalamasını alır [36].

Yeni Karar Tabanlı Algoritma YKTA (New Decision-Based Algorithm NBDA), her satır ve sütunun en küçükten en büyüğe doğru sıralanmasını ve pencerenin ortasındaki değerini medyan olarak alınmasını önerir [37].

Aşamalı Geçişli Medyan Filtre AGMF (Progressive Switching Median Filter PSMF) iki aşamalı yinelemeli bir yöntemdir [38]. İlk adımda gürültü tespiti gerçekleştirilir. İşlenen pikseller, bir sonraki yinelemede diğer gürültülü piksellerin algılanmasını sağlar.

Gürültü Uyarlamalı Bulanık Anahtarlama Medyan Filtresi, GUBAMF (Noise Adaptive Fuzzy Switching Median Filter, NAFSM), yüksek yoğunluklu tuz ve biber gürültüsünü gidermek için etkili bir yöntem sunar [39]. Bu iki aşamalı filtre önce gürültü tespiti yapar ve ardından bulanık filtreleme kullanarak gürültüyü ortadan kaldırır.

Uyarlanabilir Ağırlıklı Ortalama Filtre, UAOF (Improved Adaptive Weighted Mean Filter, IAWMF) 2020 yılında yapılmış yeni çalışmalardan biridir [40]. Uyarlanabilir bir pencere içerisinde gürültüsüz piksellerin ağırlıklı ortalamasını kullanarak gürültülü piksel değerini hesaplar. Ağırlıklı ortalama kullanılan gürültüsüz piksel değerlerinin, bulunması hedeflenen gürültülü piksele uzaklığına göre hesaplanır.

Tuz ve biber gürültüsünü gidermek için Ayrıntıya Duyarlı Filtre ADF, (Salt and Pepper Noise Removal Method Based on a Detail-Aware Filter, DAF) 2021 yılında yayımlanmıştır [41]. Öncelikle gürültülü ve gürültüsüz pikseli ayırır, sonrasında gürültülü pikseller için medyan kullanarak bir ön filtreleme işlemi yapar. Yerel olmayan ölçekler ve yoğunluk benzerliği, piksel değeri yerine piksel parçaları arasında hesaplanır. İkinci dereceden tahmine dayalı norm görüntü parçasının uzamsal yakınlığını ölçmek için kullanılır. Yerel olmayan ölçekler, ikili filtreleme ve uyarlamalı normu birleştirerek değeri hesaplar.

Tuz ve Biber gürültüsünü gidermek için derin öğrenme modeli olan Derin Evrişimli Sinir Ağı (Deep CNN for removal of salt and pepper noise) algoritması 2019 yılında yayımlanmıştır [42]. Üç aşamalı çalışan algoritmada önce ağ mimarisi inşa edilir, ikinci olarak derin öğrenme modeli eğitilir son olarak gürültü gidericiler seçilir.

2. KULLANILAN YÖNTEM

Bu çalışmada görüntü üzerindeki gürültüleri gidermek için Yüksek Boyutlu Model Gösterilimi (YBMG) ve Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi (ÇYÇG) kullanılmıştır. YBMG yöntemi, çok değişkenli fonksiyonları ayrıştıran bir tekniktir. Bu yöntem, bir fonksiyonu sabit terimler, tek değişkenli terimler, iki değişkenli terimler ve daha fazla değişkene bağlı terimlerin toplamı olarak yeniden ifade eder. Aynı zamanda, YBMG yöntemi, çok boyutlu verileri daha az boyutlu veri kümelerine ayrıştırabilmektedir [43].

ÇYÇG ise YBMG yöntemine çok benzemekle birlikte yöntemde yer alan destek işlevlerinin varlığı sayesinde farklı bir ayrıştırım yöntemi olarak varlığını sürdürür. Hem YBMG yöntemine hem de ÇYÇG yöntemine ilişkin detaylar, tezin bu bölümünde verilmektedir. [44].

2.1 Yüksek Boyutlu Model Gösterilimi ile Fonksiyon Ayrıştırımı

$f(x_1, \dots, x_N)$, çok değişkenli bir fonksiyon olsun. $f(x_1, \dots, x_N)$ için YBMG açılımı aşağıda verildiği gibidir.

$$f(x_1, \dots, x_N) = f_0 + \sum_{i=1}^N f_i(x_i) + \sum_{\substack{i,j=1 \\ i < j}}^N f_{ij}(x_i, x_j) + \dots + f_{12\dots N}(x_1, \dots, x_N) \quad (2.1)$$

Bu açılım sonlu bir toplamı ifade etmektedir. Açılımda görülen ilk terim f_0 , sabit bir fonksiyondur. $f_i(x_i)$ terimleri, tek değişkenli fonksiyonlardır ve $\binom{N}{1}$ tanedir. $f_{ij}(x_i, x_j)$ iki değişkenli fonksiyonlardır $\binom{N}{2}$ tanedir. Benzer şekilde artarak, değişken sayısı r olan fonksiyonların sayısı da $\binom{N}{r}$ kadardır. Açılımın son terimi olan $f_{12\dots N}(x_1, \dots, x_N)$ fonksiyonu ise, $f(x_1, \dots, x_N)$ fonksiyonunun tüm değişkenlerine birden bağlı olan bir fonksiyondur. Bu fonksiyonlara YBMG terimleri denir.

YBMG terimlerini eşsiz bir şekilde belirlemek için aşağıdaki yok etme koşulları kullanılmaktadır. Bu koşullar Sobol tarafından belirlenmiştir [43].

$$\int_{a_1}^{b_1} dx_1 \dots \int_{a_N}^{b_N} dx_N W_1(x_1) \cdots W_N(x_N) f_i(x_i) = 0, \quad i = 1, \dots, N \quad (2.2)$$

Burada yer alan $W_i(x_i)$ terimleri, tek değişkenli ağırlık fonksiyonlarıdır. Uygun bir iç çarpım altında yok etme fonksiyonlarının aşağıdaki diklik koşullarına karşılık geldiği Prof. Demiralp tarafından ispatlanmıştır.

$$(f_{i_1 i_2 \dots i_k}, f_{i_1 i_2 \dots i_l}) = 0, \quad i_1 i_2 \dots i_k \neq i_1 i_2 \dots i_l, \quad k, l = 1, \dots, N \quad (2.3)$$

Diklik koşulu, YBMG açılımında yer alan herhangi iki terimin birbirine dik olması koşuludur. Diklik koşulu bir iç çarpım üzerinde tanımlanmaktadır. Orjinal fonksiyonun ve YBMG açılımından elde edilen fonksiyonların karesinin integrallenebilir olduğu kabul edilir. Her bir bağımsız değişkenin tanım aralığı $(x_i \in [a_i, b_i])$ olmak üzere, karesi integrallenebilir herhangi iki $u(x_1, \dots, x_N)$ ve $v(x_1, \dots, x_N)$ fonksiyonları için, iççarpım ve ağırlık fonksiyonları altında

$$(u, v) = \int_{a_1}^{b_1} dx_1 W_1(x_1) \dots \int_{a_N}^{b_N} dx_N W_N(x_N) u(x_1, \dots, x_N) v(x_1, \dots, x_N) \quad (2.4)$$

olarak tanımlanır.

YBMG terimlerinin belirlenmesi için ağırlık fonksiyonlarının integrali her birinin bağlı olduğu değişkenin ilgili aralığında 1 olacak şekilde seçilir.

$$\int_{a_i}^{b_i} dx_i W_i(x_i) = 1, \quad i = 1, \dots, N \quad (2.5)$$

Bu normalizasyon kriteri ve öncesinde anlatılmış olan ağırlık fonksiyonları ve diklik koşulu kullanılarak YBMG terimleri elde edilir.

f_0 sabit terimini elde etmek için, YBMG eşitliğinin (Denklem 2.1) her iki tarafı $W(x_1, \dots, x_N)$ ile çarpılır ve tüm değişkenler ilgili aralıklarında integrallenir. Bu işlem neticesinde f_0 dışındaki tüm terimler yok etme koşulundan dolayı gider ve f_0

$$f_0 = \int_{a_1}^{b_1} dx_1 \dots \int_{a_N}^{b_N} dx_N W_1(x_1) \cdots W_N(x_N) f(x_1, \dots, x_N) \quad (2.6)$$

olarak bulunur.

$f_i(x_i)$ fonksiyonunu elde etmek için ise benzer şekilde YBMG eşitliğinin her iki tarafı ağırlık fonksiyonu ile çarpılır ve integrallenir fakat bu işlem yapılırken ilgili x_i değişkenine bağlı ağırlık fonksiyonu ve integral işlemi dışlanır. Bu durumda tek değişkenli $f_i(x_i)$

$$f_i(x_i) = \int_{a_1}^{b_1} dx_1 \dots \int_{a_{i-1}}^{b_{i-1}} dx_{i-1} \int_{a_{i+1}}^{b_{i+1}} dx_{i+1} \dots \int_{a_N}^{b_N} dx_N W_1(x_1) \dots W_{i-1}(x_{i-1}) \\ \cdot W_{i+1}(x_{i+1}) \dots W_N(x_N) f(x_1, \dots, x_N) - f_0, \quad i = 1, \dots, N \quad (2.7)$$

olarak elde edilir.

İki değişkenli terimlerde yine benzer şekilde, YBMG eşitliğinin her iki tarafı, $W(x_i)$ ve $W(x_j)$ hariç diğer ağırlık fonksiyonları ile çarpılıp x_i ve x_j hariç diğer değişkenlere göre ilgili aralıklarda integrallenerek

$$f_{ij}(x_i, x_j) = \int_{a_1}^{b_1} dx_1 \dots \int_{a_{i-1}}^{b_{i-1}} dx_{i-1} \int_{a_{i+1}}^{b_{i+1}} dx_{i+1} \dots \int_{a_{j-1}}^{b_{j-1}} dx_{j-1} \int_{a_{j+1}}^{b_{j+1}} dx_{j+1} \dots \int_{a_N}^{b_N} dx_N \\ \cdot W_1(x_1) \dots W_{i-1}(x_{i-1}) W_{i+1}(x_{i+1}) \dots W_{j-1}(x_{j-1}) W_{j+1}(x_{j+1}) \dots W_N(x_N) f(x_1, \dots, x_N) \\ - f_i(x_i) - f_j(x_j) - f_0, \quad i, j = 1, \dots, N \quad (2.8)$$

şeklinde elde edilir.

Artan sayıda değişkenli terimler içeren fonksiyonlarda aynı mantıkla elde edilir. Son terim olan $f_{12\dots N}(x_1, \dots, x_N)$ fonksiyonu tüm değişkenlere birden bağlıdır. Bu terimi elde etmek için genel YBMG açılımından bu fonksiyonun kendisi çekilir.

Elde edilen sonlu terimlerin toplamı orijinal fonksiyona eşittir. Elde edilen toplam sabit fonksiyondan sonra kesilirse sabit yaklaşımdır, tek değişkenli fonksiyonlardan sonra kesilirse tek değişkenli yaklaşımdır ve iki değişkenli fonksiyonlardan sonra kesilirse iki değişkenli yaklaşımdır elde edilir. Bu yaklaşımlar aşağıdaki gibidir.

$$\begin{aligned}
s_0(x_1, \dots, x_N) &= f_0 \\
s_1(x_1, \dots, x_N) &= f_0 + \sum_{i=1}^N f_i(x_i) \\
s_2(x_1, \dots, x_N) &= f_0 + \sum_{i=1}^N f_i(x_i) + \sum_{\substack{i,j=1 \\ i < j}}^N f_{ij}(x_i, x_j) \\
&\vdots \\
s_k(x_1, \dots, x_N) &= s_{k-1}(x_1, \dots, x_N) + \sum_{\substack{i_1, \dots, i_k=1 \\ i_1 < \dots < i_k}}^N f_{i_1 \dots i_k}(x_{i_1}, \dots, x_{i_k}) \\
1 \leq k \leq N
\end{aligned} \tag{2.9}$$

Orijinal fonsiyon yukarıdaki yaklaşımları kullanarak

$$f(x_1, \dots, x_N) \approx s_k(x_1, \dots, x_N) \tag{2.10}$$

olarak elde edilir.

2.2 Yüksek Boyutlu Model Gösterilimi ile Veri Ayrıştırımı

YBMG analitik yapısı bilinmeyen $x_1, \dots, x_N$ değişkenlerine bağlı, Öklit uzayının ayrık sonlu sayıda noktasında aldığı değerleri bilinen fonksiyonlara, yani verilere de uygulanabilir [45]. Her bir x_i değişkeninin tanım kümesi $\mathcal{D}_i$ olarak tanımlanır.

$$\mathcal{D}_i = \left\{ \xi_i^{(1)}, \dots, \xi_i^{(k_i)}, \dots, \xi_i^{(n_i)} \right\}, \quad i = 1, \dots, N \tag{2.11}$$

Verinin tanım kümesi, her bir x_i ye ait $\mathcal{D}_i$ lerin kartezyen çarpımından oluşan

$$\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \dots \times \mathcal{D}_N \tag{2.12}$$

kümesidir. Bu $\mathcal{D}$ kümesi üzerinde tanımlanmış fonksiyonun aldığı değerler, veriyi oluşturan değerlerdir.

Sonlu bir verinin YBMG açılımını oluştururken ağırlık fonksiyonları Dirac delta fonksiyonlarının lineer bir kombinasyonu olarak aşağıdaki şekilde tanımlanabilir.

$$W_i(x_i) = \sum_{k_i=1}^{n_i} \alpha_{k_i}^{(i)} \delta(x_i - \xi_i^{(k_i)}), \quad x_i \in [a_i, b_i], i = 1, \dots, N \tag{2.13}$$

Ağırlık fonksiyonlarının çarpımı aşağıdaki şekildedir.

$$\prod_{i=1}^N \left(\sum_{k_i=1}^{n_i} \alpha_{k_i}^{(i)} \delta(x_i - \xi_i^{(k_i)}) \right) = \prod_{i=1}^N \alpha_{k_i}^{(i)}, \quad i = 1, \dots, N \quad (2.14)$$

Bağılı olduğu değişkenin ilgili aralığında ağırlık fonksiyonunun integrali şu şekilde hesaplanır.

$$\int_{a_i}^{b_i} dx_i \sum_{k_i=1}^{n_i} \alpha_{k_i}^{(i)} \delta(x_i - \xi_i^{(k_i)}) = \sum_{k_i=1}^{n_i} \alpha_{k_i}^{(i)}, \quad i = 1, \dots, N \quad (2.15)$$

Normalizasyon koşulu her bir ağırlık fonksiyonuna uygulandığında

$$\sum_{k_i=1}^{n_i} \alpha_{k_i}^{(i)} = 1, \quad i = 1, \dots, N \quad (2.16)$$

elde edilir.

2.6'de verilen f_0 fonksiyonunu elde etmek için kullanılan ağırlık fonksiyonları yeni ağırlık fonksiyonlarıyla değiştirildiği zaman f_0 sabit fonksiyonun yapısı

$$f_0 = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \dots \sum_{k_N=1}^{n_N} \left(\prod_{i=1}^N \alpha_{k_i}^{(i)} \right) f(\xi_1^{(k_1)}, \dots, \xi_N^{(k_N)}) \quad (2.17)$$

olarak elde edilir.

Benzer şekilde veriler için elde edilen ağırlık fonksiyonları kullanılarak, $f_i(x_i)$ tek değişkenli fonksiyonların yapısı

$$f_i(\xi_i^{(k_i)}) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \dots \sum_{k_{i-1}=1}^{n_{i-1}} \sum_{k_{i+1}=1}^{n_{i+1}} \dots \sum_{k_N=1}^{n_N} \left(\prod_{\substack{m=1 \\ m \neq i}}^N \alpha_{k_m}^{(m)} \right) f(\xi_1^{(k_1)}, \dots, \xi_N^{(k_N)}) - f_0$$

$$; \xi_i^{(k_i)} \in \mathcal{D}_i ; k_i = 1, \dots, n_i ; i = 1, \dots, N \quad (2.18)$$

olarak bulunur.

Burada bulunan $f_i(x_i)$ tek değişkenli fonksiyonların yapısı n_i tane değer içeren vektör şeklindedir. Toplam $\binom{N}{1}$ kadar vektör vardır.

İki değişkenli fonksiyonların yapısı da benzer şekilde 2.8 denkleminde bulunan ağırlık fonksiyonlarını veriler için bulunan ağırlık fonksiyonlarıyla değiştirip integrallenerek

$$f_{ij}(\xi_i^{(k_i)}, \xi_j^{(k_j)}) = \sum_{k_1=1}^{n_1} \dots \sum_{k_{i-1}=1}^{n_{i-1}} \sum_{k_{i+1}=1}^{n_{i+1}} \dots \sum_{k_{j-1}=1}^{n_{j-1}} \sum_{k_{j+1}=1}^{n_{j+1}} \dots \sum_{k_N=1}^{n_N} \left(\prod_{\substack{m=1 \\ m \neq i \\ m \neq j}}^N \alpha_{k_m}^{(m)} \right) f(\xi_1^{(k_1)}, \dots, \xi_N^{(k_N)}) \\ - f_i(\xi_i^{(k_i)}) - f_j(\xi_j^{(k_j)}) - f_0 \\ ; \xi_i^{(k_i)} \in \mathcal{D}_i ; \xi_j^{(k_j)} \in \mathcal{D}_j ; k_i = 1, \dots, n_i ; k_j = 1, \dots, n_j ; i, j = 1, \dots, N \quad (2.19)$$

olarak bulunur. $f_{ij}(x_i, x_j)$ iki değişkenli terimleri x_i ve x_j değişkenlerine bağlı $n_i \times n_j$ boyutlu matris şeklindedir. her biri matris şeklindedir. Toplam $\binom{N}{2}$ tane matris vardır.

İkiden daha fazla sayıda bilinmeyen içeren terimler de benzer şekilde bulunur. YBMG terimlerinin tamamının toplamı ile orijinal veriye, terimlerinin bazılarının toplamı ile de veriye yaklaşık olarak ulaşılabilir. YBMG ile veriye yaklaştırım analitik fonksiyonlardaki gibi yapılıdır.

2.3 Yüksek Boyutlu Model Gösteriliminin Görüntüye Uygulanması

RGB formatındaki dijital görüntüler üç boyutlu dizi yapısındadır ve ayrık değerlerden oluşur. Bu nedenle YBMG veri ayrıştırımı görüntüler üzerine uygulanabilir [46].

Dijital bir görüntü *düşey piksel sayısı* $\times$ *yatay piksel sayısı* bir alanı ifade eder. Alan içerisindeki herhangi bir piksel değeri RGB (kırmızı, yeşil, mavi) yoğunluğu ifade eden, $[0,255]$ aralığındaki tam sayılardan oluşan üç değer ile belirlenir. Siyah beyaz görüntülerde RGB yapısındaki üç tamsayı değeri birbirinin aynısıdır.

Dijital bir görüntü üç değişkenli bir veri olarak ele alınır. Bu değişkenler *düşey*, *yatay* ve *renk* değişkenleridir. Bu durumda ayrık noktalarda aldığı değerler bilinen üç değişkenli bir fonksiyon gibi düşünülüp YBMG uygulanır. n_1 : *düşey piksel sayısı*, n_2 : *yatay piksel sayısı* ve n_3 : *renk sayısı* ($= 3$) şeklindedir. Görüntü değerleri ise $f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)})$, $k_1 = 1, \dots, n_1$; $k_2 = 1, \dots, n_2$; $k_3 = 1, 2, 3$ şeklindedir. $\alpha_{k_i}^{(i)}$ ağırlık fonksiyonlarının sağlaması gereken normalizasyon kriteri

$$\sum_{k_i=1}^{n_i} \alpha_{k_i}^{(i)} = 1, i = 1, 2, 3 \quad (2.20)$$

şeklindedir. Bu durumda, $\alpha_{k_i}^{(i)}$ ağırlıkları

$$\alpha_{k_i}^{(i)} = \frac{1}{n_i}, \quad i = 1, 2, 3 \quad (2.21)$$

şeklinde homojen olarak seçilebilir. 2.21 denkleminde verilen ağırlık fonksiyonları 2.17, 2.18 ve 2.19 denklemlerinde kullanılarak görüntüler için YBMG açılımının sabit terimi olan f_0

$$f_0 = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \sum_{k_3=1}^3 \frac{1}{n_1} \cdot \frac{1}{n_2} \cdot \frac{1}{3} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) \quad (2.22)$$

şeklinde, YBMG açılımının tek değişkenli terimleri f_1, f_2 ve f_3

$$f_1(\xi_1^{(k_1)}) = \sum_{k_2=1}^{n_2} \sum_{k_3=1}^3 \frac{1}{n_2} \cdot \frac{1}{3} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) - f_0 \quad (2.23)$$

$$f_2(\xi_2^{(k_2)}) = \sum_{k_1=1}^{n_1} \sum_{k_3=1}^3 \frac{1}{n_1} \cdot \frac{1}{3} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) - f_0 \quad (2.24)$$

$$f_3(\xi_3^{(k_3)}) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \frac{1}{n_1} \cdot \frac{1}{n_2} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) - f_0 \quad (2.25)$$

şeklinde, YBMG açılımının iki değişkenli terimleri f_{12}, f_{13} ve f_{23}

$$f_{12}(\xi_1^{(k_1)}, \xi_2^{(k_2)}) = \sum_{k_3=1}^3 \frac{1}{3} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) - f_1(\xi_1^{(k_1)}) - f_2(\xi_2^{(k_2)}) - f_0 \quad (2.26)$$

$$f_{13}(\xi_1^{(k_1)}, \xi_3^{(k_3)}) = \sum_{k_2=1}^{n_2} \frac{1}{n_2} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) - f_1(\xi_1^{(k_1)}) - f_3(\xi_3^{(k_3)}) - f_0 \quad (2.27)$$

$$f_{23}(\xi_2^{(k_2)}, \xi_3^{(k_3)}) = \sum_{k_1=1}^{n_1} \frac{1}{n_1} \cdot f(\xi_1^{(k_1)}, \xi_2^{(k_2)}, \xi_3^{(k_3)}) - f_2(\xi_2^{(k_2)}) - f_3(\xi_3^{(k_3)}) - f_0 \quad (2.28)$$

şeklinde elde edilir.

Bu terimler kullanılarak görüntüye sabit yaklaştırım, tek değişkenli yaklaştırım ve iki değişkenli yaklaştırım yapılabilir.

Bu tez çalışmasında, görüntü işleme alanındaki gürültü giderme sorununa odaklanılmaktadır. Yüksek Boyutlu Model Gösterilimi (YBMG) terimleri kullanılarak yeni bir filtre tasarlanacaktır. Bu filtre, görüntüdeki istenmeyen gürültüleri azaltmayı amaçlamaktadır. YBMG terimleri, görüntü verilerine belli oranlarda yaklaşma imkanı sunar ve eksik verilerin yeniden hesaplanmasında kullanılır. Bu yaklaşımlar, görüntüdeki gürültüyü oluşturan eksik verilerin düzeltilmesini sağlayacaktır. Çalışma boyunca, YBMG terimleri detaylı bir şekilde incelenecek ve yeni filtre tasarımının performansı değerlendirilecektir.

2.4 Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi ile Fonksiyon Ayrıştırımı

$f(x_1, \dots, x_N)$, çok değişkenli bir fonksiyon olsun. $f(x_1, \dots, x_N)$ için ÇYÇG açılımı aşağıda verildiği gibidir [47].

$$\begin{aligned} f(x_1, \dots, x_N) &= f_0 \prod_{j=1}^N s_j(x_j) + \sum_{i=1}^N f_i(x_i) \prod_{\substack{j=1 \\ j \neq i}}^N s_j(x_j) \\ &+ \sum_{\substack{i_1, i_2=1 \\ i_1 < i_2}}^N f_{i_1, i_2}(x_{i_1}, x_{i_2}) \prod_{\substack{j=1 \\ j \neq i_1, i_2}}^N s_j(x_j) \\ &+ \dots + f_{12 \dots N}(x_1, x_2, \dots, x_N) \end{aligned} \quad (2.29)$$

Bu açılım sonlu bir toplamdır. Terimler YBMG ile benzerlik göstermektedir. Açılımda görülen ilk terim f_0 , sabit bir fonksiyondur. $f_i(x_i)$ terimleri, tek değişkenli fonksiyonlardır. $f_{ij}(x_i, x_j)$ iki değişkenli fonksiyonlardır. $f_{12 \dots N}(x_1, \dots, x_N)$ fonksiyonu ise, $f(x_1, \dots, x_N)$ fonksiyonunun tüm değişkenlerine birden bağlı olan bir fonksiyondur. Bu fonksiyonlara ÇYÇG terimleri denir. YBMG'den farklı olarak $s_j(x_j)$ destek fonksiyonları vardır. Destek fonksiyonlarının karelerinin verilen ağırlık fonksiyonları ile ilgili aralıktaki integralleri 1 olmalıdır.

$$\int_{a_i}^{b_i} dx_i W_i(x_i) s_i(x_i)^2 = 1 \quad 1 \leq i \leq N \quad (2.30)$$

ÇYÇG'nin sabit terimini elde etmek için aşağıdaki operatör tanımlanmıştır.

$$\mathcal{J}_0 f(x_1, \dots, x_N) \equiv \int_{a_1}^{b_1} dx_1 W_1(x_1) \dots \times \int_{a_N}^{b_N} dx_N W_N(x_N) \prod_{j=1}^N s_j(x_j) f(x_1, \dots, x_N) \quad (2.31)$$

ÇYÇG'nin tek değişkenli terimlerini elde etmek için

$$\begin{aligned} \mathcal{J}_i f(x_1, \dots, x_N) &\equiv \int_{a_1}^{b_1} dx_1 W_1(x_1) \dots \int_{a_{i-1}}^{b_{i-1}} dx_{i-1} W_{i-1}(x_{i-1}) \\ &\int_{a_{i+1}}^{b_{i+1}} dx_{i+1} W_{i+1}(x_{i+1}) \dots \int_{a_N}^{b_N} dx_N W_N(x_N) \prod_{\substack{j=1 \\ j \neq i}}^N s_j(x_j) f(x_1, \dots, x_N) \end{aligned} \quad (2.32)$$

operatörü tanımlanmıştır. Bahsedilen integralleri hesaplamak için bazı varsayımlar kullanılmıştır. Bunlar yok etme koşulları ve normalizasyon kriteridir [48].

Yok etme koşulları

$$\int_{a_{i_s}}^{b_{i_s}} dx_{i_s} W_{i_s}(x_{i_s}) s_{i_s}(x_{i_s}) f_{i_1 \dots i_N}(x_{i_1}, \dots, x_{i_N}) = 0 \quad (2.33)$$

normalizasyon kriteri

$$\int_{a_i}^{b_i} dx_i W_i(x_i) = 1, \quad 1 \leq i \leq N \quad (2.34)$$

şeklindedir. Bu koşullar altında ÇYÇG'nin sabit terimi

$$f_0 = \int_{a_1}^{b_1} dx_1 W_1(x_1) \dots \int_{a_N}^{b_N} dx_N W_N(x_N) \prod_{j=1}^N s_j(x_j) f(x_1, \dots, x_N) \quad (2.35)$$

şeklinde, tek değişkenli terimleri

$$f_i(x_i) = \int_{a_1}^{b_1} dx_1 W_1(x_1) \dots \int_{a_{i-1}}^{b_{i-1}} dx_{i-1} W_{i-1}(x_{i-1}) \int_{a_{i+1}}^{b_{i+1}} dx_{i+1} W_{i+1}(x_{i+1}) \dots \int_{a_N}^{b_N} dx_N W_N(x_N) \prod_{\substack{j=1 \\ j \neq i}}^N s_j(x_j) f(x_1, \dots, x_N) - f_0 s_i(x_i) \quad (2.36)$$

şeklinde elde edilir. Sabit terim ve tek değişkenli terimler kullanılarak orijinal fonksiyona sabit yaklaştırım ve tek değişkenli yaklaştırım sırasıyla aşağıdaki şekillerde yazılır [49].

$$S_0^{(e)}(x_1, \dots, x_N) = f_0 \prod_{j=1}^N s_j(x_j)$$

$$S_1^{(e)}(x_1, \dots, x_N) = S_0^{(e)}(x_1, \dots, x_N) + \sum_{i=1}^N f_i(x_i) \prod_{\substack{j=1 \\ j \neq i}}^N s_j(x_j)$$

ÇYÇG açılımında önemli noktalardan biri olan destek fonksiyonlarının yapısı

$$s_j(x_j) = \frac{\kappa_j(x_j)}{\left[\int_{a_j}^{b_j} dx_j [\kappa_j(x_j)]^2 \right]^{\frac{1}{2}}}$$

şeklinde alınır ve burada

$$\kappa_j(x_j) = \prod_{\substack{i=1 \\ i \neq j}}^N \int_{a_i}^{b_i} dx_i f(x_1, \dots, x_N), \quad 1 \leq j \leq N$$

olarak alınırsa yaklaştırım için oldukça iyi sonuçlar elde edilmektedir.

2.5 Çok Değişkenliliği Yükseltmiş Çarpımlar Gösterilimi ile Veri Ayırıştırımı

ÇYÇG aynı YBMG yönteminde olduğu gibi analitik yapısı bilinmeyen, Öklit uzayının ayırık sonlu noktasında aldığı değerleri bilinen fonksiyonlara yani verilere de uygulanabilir [47]. Verinin tanım kümesi,

$$\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N \quad (2.37)$$

burada

$$\mathcal{D}_j \equiv \left\{ \xi_j^{(k_j)} \right\}_{k_j=1}^{k_j=n_j} = \left\{ \xi_j^{(1)}, \dots, \xi_j^{(n_j)} \right\}, \quad 1 \leq j \leq N \quad (2.38)$$

şeklindedir. Ağırlık fonksiyonları Dirac delta fonksiyonlarını kullanarak aşağıdaki şekilde tanımlanır.

$$W_j(x_j) \equiv \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \delta(x_j - \xi_j^{(k_j)}), \quad x_j \in [a_j, b_j], \quad 1 \leq j \leq N \quad (2.39)$$

Verilen ağırlık fonksiyonları altında,

$$\sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} = 1, \quad \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \left[s_j(\xi_j^{(k_j)}) \right]^2 = 1, \quad 1 \leq j \leq N. \quad (2.40)$$

ilişkileri elde edilir. Destek fonksiyon yapıları, seçilen ağırlık fonksiyonu ve yukarıda belirtilen koşullar dikkate alındığında veri modelleme için aşağıdaki şekilde belirlenebilir.

$$s_j(\xi_j^{(i_j)}) = \frac{\mathcal{K}_j(\xi_j^{(i_j)})}{\left[\sum_{i_j=1}^{n_j} \alpha_{i_j}^{(j)} \left[\mathcal{K}_j(\xi_j^{(i_j)}) \right]^2 \right]^{\frac{1}{2}}} \quad (2.41)$$

burada

$$\mathcal{K}_j(\xi_j^{(i_j)}) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_{j-1}=1}^{n_{j-1}} \sum_{i_{j+1}=1}^{n_{j+1}} \cdots \sum_{i_N=1}^{n_N} \left(\prod_{\substack{r=1 \\ r \neq j}}^N \alpha_{i_r}^{(r)} \right) f(\xi_1^{(i_1)}, \dots, \xi_j^{(i_j)}, \dots, \xi_N^{(i_N)}). \quad (2.42)$$

2.35 denklemindeki ağırlık ve destek fonksiyonları veriler için olan ağırlık fonksiyonlarıyla yer değiştirildiğinde ÇYÇG'nin veri için sabit terimi

$$f_0 = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_N=1}^{n_N} \left(\prod_{j=1}^N \alpha_{i_j}^{(j)} s_j(\xi_j^{(i_j)}) \right) f(\xi_1^{(i_1)}, \dots, \xi_N^{(i_N)}) \quad (2.43)$$

2.36 denklemdeki ağırlık ve destek fonksiyonları veriler için olan ağırlık fonksiyonlarıyla yer değiştirildiğinde ÇYÇG'nin veri için tek değişkenli terimleri

$$f_m(\xi_m^{(i_m)}) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_{m-1}=1}^{n_{m-1}} \sum_{i_{m+1}=1}^{n_{m+1}} \cdots \sum_{i_N=1}^{n_N} \left(\prod_{\substack{j=1 \\ j \neq m}}^N \alpha_{i_j}^{(j)} s_j(\xi_j^{(i_j)}) \right) f(\xi_1^{(i_1)}, \dots, \xi_m^{(i_m)}, \dots, \xi_N^{(i_N)}) \quad (2.44)$$

olarak elde edilir.

ÇYÇG'nin bir görüntüye uygulanması YBMG'nin görüntüye uygulanmasıyla aynı mantığa sahiptir. Dolayısı ile bir görüntü üç boyutlu bir dizi olarak düşünülebilir ve ÇYÇG algoritması bu şekilde uygulanır.





3. ÖNERİLEN YÖNTEM

Bu bölümde Yüksek Boyutlu Model Gösterilimi ve Çok Değişkenliliği Artırılmış Çarpımlar Gösterilimi kullanılarak tasarlanan filtreler yer verilmiştir. Geliştirilen bu filtreler arasında en yüksek performansa sahip olan filtreye Yüksek Boyutlu Model Gösterilimi Medyan Filtresi, YBMGMF adı verilmiştir.

Görüntü üzerinde bulunan alacalı ve gauss gürültülerini gidermeye yönelik ÇYÇG terimleri kullanılarak iki farklı algoritma oluşturulmuştur. Algoritma 1’de verilen birinci algoritmada filtreleme işlemi ÇYÇG sabit terimi kullanılarak yapılmıştır. Görüntü verisinin ilk 3x3 boyutundaki piksel değerleri ile birinci pencere matrisi oluşturulmuş, oluşturulan pencere matrisine ÇYÇG algoritması uygulanmış, algorithmadan elde edilen sabit terim f_0 , pencere matrisinin merkezindeki piksel değeri olarak atanmıştır. Pencere matrisi birer piksel kaydırılarak bütün görüntü üzerinde aynı filtreleme işleme yapılmıştır. Bu tez çalışmasından bu algorithmanda ÇYÇGMF1 (ÇYÇG medyan filtre) filtresi olarak bahsedilecektir.

```
Test görüntüsünü oku;  
Görüntüye gürültü ekle;  
Gürültü eklenen görüntünün ilk üç satır ve sütunuyla birinci pencere matrisini  
oluştur;  
for Hedefteki pencere matrisi do  
    Pencere matrisine ÇYÇG uygulayarak  $f_0$  sabit terimini hesapla;  
    Hesaplanan değeri pencere matrisinin merkezindeki piksel değeri olarak  
    ata;  
    Pencere matrisini kestiğin yere yapıştır;  
end  
Birer piksel kaydırarak pencereyi bütün görüntüde gezdir;  
return Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini döndür;
```

Algorithm 1: ÇYÇGMF1 algoritması.

Algoritma 2’de verilen ikinci algoritmada filtreleme işlemi ÇYÇG tek değişkenli terimleri kullanılarak yapılmıştır. Görüntü verisinin ilk 3x3 boyutundaki piksel değerleri ile birinci pencere matrisi oluşturulmuş, oluşturulan pencere matrisine ÇYÇG algoritması uygulanmış, algorithmadan elde edilen tek değişkenli f_{12} ve f_{13} vektörleri kullanılmıştır. Bu vektörlerin her ikisinin de ayrı ayrı medyanları hesaplanmış ve

bu medyanların ortalaması pencere matrisinin merkezindeki piksel değeri olarak atanmıştır. Pencere matrisi birer piksel kaydırılarak bütün görüntü üzerinde aynı filtreleme işleme yapılmıştır. Bu tez çalışmasında bu filtreden ÇYÇGMF2 filtresi olarak bahsedilecektir. ÇYÇGMF2 ile elde edilen sonuçlar ÇYÇG1 ile elde edilen sonuçlardan daha iyidir. Bu iki yönteme ve bu bölümde anlatılacak tüm yöntemlere ait sonuçlar bir sonraki bölümde verilmiştir.

```

Test görüntüsünü oku;
Görüntüye gürültü ekle;
Gürültü eklenen görüntünün ilk üç satır ve sütunuyla birinci pencere matrisini
oluştur;
for Hedefteki pencere matrisi do
    Pencere matrisine ÇYÇG uygulayarak tek değişkenli terimleri bul;
     $f_{12}$  ve  $f_{13}$  vektörlerinin medyanlarını hesapla;
    Bulunan medyanların ortalamasını al;
    Hesaplanan ortalamayı pencere matrisinin merkezindeki piksel değeri
    olarak ata;
    Pencere matrisini kestiğin yere yapıştır;
end
Birer piksel kaydırarak pencereyi bütün görüntüde gezdir;
return Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini döndür;

```

Algorithm 2: ÇYÇGMF2 algoritması.

Çalışmanın devamında tuz ve biber gürültüsünü gidermeye yönelik hem YBMG hem de ÇYÇG kullanılarak yeni filtreler oluşturulmuştur. Algoritma 3’de verilen yöntemde, öncelikli olarak 0, 1 yoğunlukla tuz ve biber gürültüsüne sahip görüntü verisinde piksel değeri 255 olan piksellerin tamamı 0’a dönüştürülmüştür. Buradaki amaç muhtemel gürültü olduğu düşünülen 0 ve 255 değerli piksel noktalarının belirlenmesidir. Birinci satır ve birinci sütundan başlanarak görüntü verisi üzerinde birer kayarak gezdirilir. 0 tespit edilen noktada, 0 merkezde kalacak şekilde 3x3 lük pencere matrisi oluşturulur. Devamında pencere matrisine YBMG uygulanarak, merkezdeki muhtemel gürültü olduğu düşünülen piksel değeri yerine YBMG sabit terimi f_0 konulur. Elde edilen yeni değerli pencere matrisine yeniden YBMG uygulanır. YBMG sabit ve tek değişkenli terimleri hesaplanır. Tek değişkenli terimlerin medyanlarının ortalamasına sabit terim eklenerek pencere matrisinin merkez piksel değeri olarak atanır. Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değeri hesaplanır. YBMG ile oluşturulan bu algoritmadan bu çalışmada YBMGMF1 (YBMG medyan filtre) olarak bahsedilecektir.

```

Test görüntüsünü oku;
Piksel değeri 255 olan her noktayı 0 ile değiştir;
Pikselleri birer kaydırarak gez ve 0 olan piksel değerini yakaladığın yerde
    merkez piksel 0 olacak şekilde 3x3 pencere oluştur ;
for Hedefteki pencere matrisi do
    Merkez piksel değerini  $f_0$  ile değiştir;
    YBMG uygula;
    YBMG tek değişkenli terimlerin medyanlarını hesapla;
    Bulunan medyanların ortalamasını al;
    Ortalamaya YBMG sabit terimi ekle ve elde edilen değeri merkez piksel
        olarak ata;
    Pencere matrisini kestiğin yere yapıştır;
end
return Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini döndür;

```

Algorithm 3: YBMGMF1 algoritması.

YBMGMF1 filtresinin performansının artırılması amacıyla algoritma yeniden elden geçirilerek iteratif hale getirilmiş ve YBMGMF2 olarak isimlendirilerek Algoritma 4’de verilmiştir. Test görüntüleri için optimal iterasyon sayısı deneysel olarak belirlenerek algoritma en iyi sonucu verene kadar tekrar ettirilmiştir. Bunun amacı algoritma bir kez uygulandığında giderilemeyen gürültü taşıyıcısı piksellerin diğer tekrarlarda yakalanarak ortadan kaldırılmasıdır.

```

Test görüntüsünü oku;
Piksel değeri 255 olan her noktayı 0 ile değiştir;
Pikselleri birer kaydırarak gez ve 0 olan piksel değerini yakaladığın yerde
    merkez piksel 0 olacak şekilde 3x3 pencere oluştur ;
for Hedefteki pencere matrisi do
    Merkez piksel değerini  $f_0$  ile değiştir;
    YBMG uygula;
    YBMG tek değişkenli terimlerin medyanlarını hesapla;
    Bulunan medyanların ortalamasını al;
    Ortalamaya YBMG sabit terimi ekle ve elde edilen değeri merkez piksel
        olarak ata;
    Pencere matrisini kestiğin yere yapıştır;
end
Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini hesapla;
Elde edilen görüntüyü, girdi görüntüsü olarak alarak, PSNR değeri
    değişmeyene kadar algoritmayı tekrar et;
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;

```

Algorithm 4: YBMGMF2 algoritması.

YBMGMF3 olarak bahsedeceğimiz algortmada ise YBMGMF1 filtresinden farklı olarak 255 olan piksel değerleri 0’a dönüştürülmeden aynen korunur. Birinci satır

ve birinci sütundan başlanarak görüntü verisi üzerinde dolaşılırken değeri 0 veya 255 olarak belirlenen pikseller merkezde kalacak şekilde etrafında 3x3'lük pencere matrisi oluşturulur. YBMGMF3 filtresi Algoritma 5'de verilmiştir. Benzer şekilde YBMGMF4 olarak bahsedeceğimiz algorithmada ise YBMGMF3 filtresinin iteratif olarak gürültülü piksel değeri kalmayınca kadar tekrar edilmesi ile oluşturulan filtredir ve algoritması Algoritma 6'da verilmiştir.

```

Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde kalacak şekilde 3x3 boyutlu pencere oluştur;
for Hedefteki pencere matrisi do
    Merkez piksel değerini  $f_0$  ile değiştir;
    YBMG uygula;
    YBMG tek değişkenli terimlerin medyanlarını hesapla;
    Bulunan medyanların ortalamasını al;
    Ortalamaya YBMG sabit terimi ekle ve elde edilen değeri merkez piksel
    olarak ata;
    Pencere matrisini kestiğin yere yapıştır;
end
return Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini döndür;

```

Algorithm 5: YBMGMF3 algoritması.

```

Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde kalacak şekilde 3x3 boyutlu pencere oluştur;
for Hedefteki pencere matrisi do
    Merkez piksel değerini  $f_0$  ile değiştir;
    YBMG uygula;
    YBMG tek değişkenli terimlerin medyanlarını hesapla;
    Bulunan medyanların ortalamasını al;
    Ortalamaya YBMG sabit terimi ekle ve elde edilen değeri merkez piksel
    olarak ata;
    Pencere matrisini kestiğin yere yapıştır;
end
Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini hesapla;
Elde edilen görüntüyü, girdi görüntüsü olarak alarak, PSNR değeri
değişmeye kadar algoritmayı tekrar et;
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;

```

Algorithm 6: YBMGMF4 algoritması.

Tuz ve biber gürültüsü içeren görüntü verisine YBMG uygulandığında elde edilen iki değişkenli terimlerin gürültüyü tutma tahmininden yola çıkarak da bir algoritma geliştirilmiştir. Bu algorithmada öncelikle yine 255 olan piksel değerleri yerlerini

belirlemek amacıyla 0'a dönüştürülmüş, devamında görüntü üzerinde gezerken 0 tespit edilen pikseller merkezde kalacak şekilde 3×3 'lük pencere matrisi oluşturulmuştur. Oluşturulan matrise YBMG uygulanarak, YBMG terimlerinden iki bileşenli olanlar çıkarılarak pencere matrisi tekrar elde edilip, görüntüde aynı lokasyonuna tekrar yerleştirilmiştir. Fakat bu denemede elde edilen sonuçlar medyan filtrenin gerisinde kalmıştır. Aynı yöntem benzer şekilde iteratif olarak uygulanmış, yani algoritma bir kez uygulandıktan sonra gürültülü piksellerin belirlenememesi ihtimaliyle deneysel olarak elde edilen uygun sayılarda tekrar edilmiştir. Sonuçta ise yine medyan filtre ile kıyasla daha geride PSNR değerleri elde edilmiştir.

İki değişkenli terimleri dışlayarak tasarlanan algoritma 255 ve 0 olan piksel değerlerinin tamamı korunarak tekrar denenmiş fakat daha iyi bir sonuç elde edilememiştir. Bir önceki anlatılan iteratif olan algoritmaya benzer şekilde yine 255 ve 0 olan piksel değerleri korunarak, optimal sayıda iki değişkenli terimler dışlanarak YBMG uygulanmış fakat yine medyan filtreden daha düşük PSNR değerleri elde edilmiştir. Bu nedenle bu algoritmaların detaylarına yer verilmemiştir.

İteratif yöntemlerden farklı olarak tek seferde gürültülü bütün piksel noktalarını bulmaya yönelik geliştirilen YBMGMF5 olarak bahsedilecek olan yöntem Algoritma 7'de verilmiştir. Burada pikseller üzerinde gezilirken bulunan 0 veya 255 olan noktalar merkezde kalacak şekilde etrafına 3×3 lük pencereler oluşturulmuştur. Fakat oluşan pencerede merkez piksel dışındaki noktalarda da 0 veya 255 değerleri yani gürültülü noktalar olma tahmininden yola çıkılarak pencere matrisindeki bulunabilen bütün 0 ve 255 değerleri f_0 bileşeni ile değiştirilmiştir. Buradaki amaç tek seferde bütün gürültülü noktaların belirlenmesidir. Devamında elde edilen yeni değerli pencere matrisine YBMG uygulanır. YBMG sabit ve tek değişkenli terimleri hesaplanır. Tek değişkenli terimlerin medyanlarının ortalamasına sabit terim eklenerek pencere matrisinin merkez piksel değeri olarak atanır. Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değeri hesaplanır.

YBMG ile yapılan çalışmaları geliştirmek için piksel değerlerindeki geçişin ani ve hızlı olduğu yerlerde daha doğru sonuç elde etmek amacıyla hibrit bir model oluşturulmuştur. Bu yeni algoritma bu çalışmada YBMGMF6 olarak adlandırılmıştır. YBMGMF6 algoritmasında gürültülü olduğu düşünülen 0 veya 255 değerlerini merkez piksel kabul ederek etrafı 3×3 boyutunda pencere olarak alınmıştır. Pencerenin

```

Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde kalacak şekilde 3x3 boyutlu pencere oluştur;
for Hedefteki pencere matrisi do
    Bulunan bütün 0 ve 255 noktalarını  $f_0$  ile değiştir;
    YBMG uygula;
    YBMG tek değişkenli terimlerin medyanlarını hesapla;
    Bulunan medyanların ortalamasını al;
    Ortalamaya YBMG sabit terimi ekle ve elde edilen değeri yeri belirlenen
    bütün 0 ve 255 noktalarına ata;
    Pencere matrisini kestiğin yere yapıştır;
end
return Girdi görüntüsü ile çıktı görüntüsü arasındaki PSNR değerini döndür;

```

Algorithm 7: YBMGMF5 algoritması.

standart sapması hesaplanarak standart sapmanın deneysel olarak elde edilen eşik değerinden küçük olduğu yerlerde YBMG sabit ve birli terimleri ile elde edilen değer kullanılmış, standart sapmanın eşik değerinden büyük olduğu yerlerde tekrarsız medyan filtre kullanılmıştır. Algoritma iteratif olarak uygulanarak görüntü iyileştirilmiştir. YBMGMF6 filtresi Algoritma 8’de verilmiştir.

```

Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;
for Hedefteki pencere matrisi do
    Pencere içindeki gürültüsüz piksellerin ortalamasını ve standart sapmasını
    hesapla;
    Merkez piksel değerini bulunan ortalama değer ile değiştir;
    Standart sapma belirlenen eşik değerinin üstündeyse, pencereye tekrarsız
    medyan uygulayarak merkez piksel değerini belirle, değilse YBMG ile
    elde edilen tek değişkenli terimlerin medyanlarının ortalamasına sabit
    terimi ekleyerek bulunan değeri merkez piksel değeri olarak belirle;
    Pencere matrisini kestiğin yere yapıştır;
end
Elde edilen görüntüyü, girdi görüntüsü olarak alarak, algoritmayı iteratif
olarak tekrar et;
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;

```

Algorithm 8: YBMGMF6 algoritması.

Yapılan hibrit model düşük yoğunlukta önceden var olan sonuçlarda iyileşme sağlarken yüksek yoğunlukta başarılı sonuçlar verememiştir. Yüksek yoğunlukta başarı elde eden yöntemler incelendiğinde küçük pencere boyutunun yeterli olmadığı görülerek, pencere boyutu uyarlamalı hale getirilmiştir. Çok yüksek yoğunlukta tuz ve

biber gürültüsünde 3x3 pencerelerde çok az sayıda orijinal değerin bulunması ya da hiç bulunmaması, gerekli değeri bulmak için yeterli olmadığı için, pencere boyutu büyütülmüştür. Yoğunluk yükseldikçe pencere boyutu artırıldığında daha iyi sonuçlar elde edilmekte, görüntü gürültülerden yüksek yoğunlukta da arındırılmakta fakat bununla beraber hesaplama karmaşıklığı arttığı için hesaplama zamanı da artmaktadır.

Bu problemi aşabilmek amacıyla tuz ve biber gürültüsünü YBMG kullanarak giderebilmek için farklı bir algoritma arayışına gidilmiştir ve pencerelerin simetrisinden faydalanacak şekilde yeni bir algoritma geliştirilmiştir. Bu çalışmada YBMGMF7 olarak adlandırılan bu algoritmada, 3x3 boyutunda alınan pencereler, sağ, sol, yukarı, aşağı ve çapraz konumlara aynalanarak 9x9 boyutuna çıkarılmıştır. Elde edilen yeni pencerenin gürültülü olmayan piksellerinin ortalaması hesaplanarak, gürültülü pikseller yerine atanması sonrasında YBMG ile tasarlanan gürültü temizleme algoritması bu pencereye uygulanarak, merkezdeki piksel değeri hesaplanmıştır. YBMGMF7 algoritması Algoritma 9’de verilmiştir.

```
Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;
for Hedefteki pencere matrisi do
    3x3 pencerenin simetrisini alarak boyutunu 9x9 olarak büyüt;
    Pencere içindeki gürültüsüz piksellerin ortalamasını hesapla;
    Gürültülü piksel değerlerini bulunan ortalama değer ile değiştir;
    Elde edilen yeni matriste, YBMG ile elde edilen tek değişkenli terimlerin
    medyanlarının ortalamasına sabit terimi ekleyerek bulunan değeri merkez
    piksel değeri olarak belirle;
    Pencere matrisini kestiğin yere yapıştır;
end
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;
```

Algorithm 9: YBMGMF7 algoritması.

YBMGMF6 ve YBMGMF7 de verilen yöntemler birleştirilerek YBMGMF8 olarak adlandırılan yeni bir yöntem elde edilmiştir. Bu yöntemde öncelikle gürültülü değeri merkez kabul eden 3x3 boyutundaki pencereler simetrisi yardımıyla 9x9 boyutuna çıkarılmış, devamında gürültüsüz piksellerin ortalaması ve standart sapması hesaplanmıştır. Gürültülü piksel değerleri bulunan ortalama ile değiştirildikten sonra, önceden bulunan standart sapma, optimal eşik değerinin üzerindeyse pencereye tekrarsız medyan uygulayarak, değilse YBMG ile oluşturulan gürültü

giderme algoritması uygulayarak penceredeki merkez piksel değeri hesaplanmıştır. YBMGMF8 yöntemi Algoritma 10'de verilmiştir.

```
Test görüntüsünü oku;  
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın  
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;  
for Hedefteki pencere matrisi do  
    3x3 pencerenin simetrilerini alarak boyutunu 9x9 olarak büyüt;  
    Pencere içindeki gürültüsüz piksellerin ortalamasını ve standart sapmasını  
    hesapla;  
    Gürültülü piksel değerlerini bulunan ortalama değer ile değiştir;  
    Elde edilen yeni matriste standart sapma belirlenen eşik değerinin  
    üstündeyse tekrarsız medyan uygulayarak merkez piksel değerini belirle,  
    değilse YBMG ile elde edilen tek değişkenli terimlerin medyanlarının  
    ortalamasına sabit terimi ekleyerek bulunan değeri merkez piksel değeri  
    olarak belirle;  
    Pencere matrisini kestiğin yere yapıştır;  
end  
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;
```

Algorithm 10: YBMGMF8 algoritması.

Yüksek Boyutlu Model Gösterilimi ile geliştirilen filtrelerden ilham alınarak Çok Değişkenliliği Yükseltmiş Çarpımlar Gösterilimi (ÇYÇG) ile de tuz ve biber gürültüsünü gidermeye yönelik yeni bir algoritma daha üretilmiştir. Algoritmanın üretilme nedeni YBMG ile geliştirilen algoritmadan daha iyi çalışabilecek bir algoritma bulma çabasıdır. Bu amaçla YBMG ile oluşturulan ve en iyi sonuç veren algoritma Çok Değişkenliliği Yükseltmiş Çarpımlar Gösterilimi (ÇYÇG) ile kullanılacak şekilde geliştirilmiştir ve bu yöntem bu çalışmada ÇYÇGMF3 olarak adlandırılmıştır. Siyah beyaz görüntü üzerinde gezilirken gürültü olarak varsayılan 0 veya 255 değerleri bulunduğu yerde, bu değerler merkez piksel olarak kabul edilerek 3x3 lük pencereler oluşturulmuştur. Bu pencerelerin sağ, sol, yukarı, aşağı ve çapraz bölgelere göre simetrileri alınarak 9x9 boyutuna çıkarılmıştır. Pencere içindeki gürültüsüz piksellerin ortalaması alınarak gürültülü pikseller yerine konulmuştur. Elde edilen yeni matrise ÇYÇG algoritması uygulanarak ÇYÇG'nin tek değişkenli terimlerinin medyanlarının ortalamasına sabit terim eklenerek merkez piksel olarak atanmıştır. ÇYÇGMF3 yöntemi Algoritma 11'de verilmiştir.

ÇYÇGMF3'de tasarlanan Algoritma, YBMGMF7'de tasarlanan algoritma kadar iyi sonuç vermemiştir. Bunu iyileştirmek için YBMG ve ÇYÇG yöntemleri

```

Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;
for Hedefteki pencere matrisi do
    3x3 pencerenin simetrilerini alarak boyutunu 9x9 olarak büyüt;
    Pencere içindeki gürültüsüz piksellerin ortalamasını;
    Gürültülü piksel değerlerini bulunan ortalama değer ile değiştir;
    Elde edilen yeni matriste, ÇYÇG ile elde edilen tek değişkenli terimlerin
    medyanlarının ortalamasına sabit terimi ekleyerek bulunan değeri merkez
    piksel değeri olarak belirle;
    Pencere matrisini kestiğin yere yapıştır;
end
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;

```

Algorithm 11: ÇYÇGMF3 algoritması.

birlikte kullanılarak bir yöntem daha geliştirilmiş ve bu yöntem ÇYÇGMF4 olarak adlandırılmıştır. ÇYÇGMF4 yönteminde YBMGMF7 ile bir kere filtrelenmiş görüntü kullanılarak ÇYÇG için gerekli olan destek fonksiyonları elde edilmiştir. Devamında bu destek fonksiyonları kullanılarak ÇYÇG terimleri bulunmuş ve tek değişkenli terimlerin medyanlarının ortalamasına sabit terim eklenerek pencerelerdeki merkez piksel değeri hesaplanmıştır. ÇYÇGMF4 olarak adlandırılan bu yöntem Algoritma 12’de verilmiştir.

```

Test görüntüsünü oku;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;
Görüntüyü YBMGMF7 algoritması ile filtrele;
for Hedefteki pencere matrisi do
    3x3 pencerenin simetrilerini alarak boyutunu 9x9 olarak büyüt;
    Pencere içindeki gürültüsüz piksellerin ortalamasını hesapla;
    Gürültülü piksel değerlerini bulunan ortalama değer ile değiştir;
    Elde edilen yeni matriste, ÇYÇG için gerekli olan destek fonksiyonlarını
    YBMGMF7 ile filtrelenmiş görüntüyü kullanarak elde et ve bu destek
    fonksiyonlarını kullanarak ÇYÇG’nin tek değişkenli terimlerin
    medyanlarının ortalamasına sabit terimi ekleyerek bulunan değeri merkez
    piksel değeri olarak belirle;
    Pencere matrisini kestiğin yere yapıştır;
end
return İlk görüntü ile son görüntü arasındaki PSNR değerini döndür;

```

Algorithm 12: ÇYÇGMF4 algoritması.

Buraya kadar yapılan çalışmalarda Yüksek Boyutlu Model Gösterilimi kullanılarak geliştirilen filtrede düşük yoğunlukta çok iyi sonuçlar elde edilmiş fakat yoğunluk

yükseldikçe filtre çok iyi sonuç vermemiştir. Çalışmanın devamında yüksek yoğunlukta da algoritmanın verimliliği artırılmıştır. Yapılan çalışmalar yoğunluk yükseldikçe pencere boyutu 3×3 kaldığında, pencere içerisinde gürültüsüz piksel bulunamadığı için algoritmanın belli bir yoğunluğun üzerinde çalışmadığını göstermiştir. Yüksek yoğunlukta da iyi sonuç elde etmek için filtredeki pencere boyutu uyarlamalı hale getirilmiştir. Böylelikle çalışılan pencere boyutu küçük olduğunda bulunamayan orijinal piksel değeri pencere boyutu büyüdüğünde bulunabilecektir. Bunun için öncelikle görüntünün tamamında sağa, sola, yukarı, aşağı ve köşelere üçer birimlik piksel değerlerinin simetrileri kopyalanarak görüntü boyutu artırılmıştır. Böylelikle ilk pikselin gürültülü olması durumunda önce çevresinde 3×3 oluşturulan pencerede orijinal değer olup olmadığına bakılmış, eğer yoksa pencere boyutu 5×5 boyutuna çıkarılmış burada da gürültüsüz piksel olmaması durumunda boyut 7×7 olarak büyütülmüştür. Devamında pencere kaydırılarak bütün görüntü üzerinde aynı algoritma gezdirilmiştir. Üçer birimlik genişletme işlemi ilk pikselden son piksele kadar görüntünün tamamını 7×7 pencere boyutuna elverişli hale getirmiştir. Böylelikle çok yüksek yoğunlukta da orijinal piksel değeri yakalanarak algoritma çok iyi sonuç verir hale gelmiştir. Algoritma bir kez uygulandıktan sonra yakalanamayan gürültülü piksel kalması ihtimali için temizlenen görüntü birer piksellik simetrileri alınarak 3×3 boyutlu pencere için üç kez daha çalıştırılmıştır. Bu hali ile bütün gürültülü piksel değerleri çok yüksek yoğunluk değerlerinde bile yakalanarak giderilmiştir. Güncellenen bu algoritmadan YBMGMF9 olarak bahsedilecek olup, YBMGMF9 algoritması Algoritma 13'de verilmiştir.

Çok Değişkenliliği Yükseltilmiş Çarpımlar Gösterilimi ile buraya kadar yapılan algoritmalarda düşük yoğunlukta tuz ve biber gürültüsü giderilmiş fakat YBMG ile üretilen algoritma daha iyi sonuç vermiştir. ÇYÇG ile yapılan çalışmalara ek olarak öncelikle gürültülü piksel değerini hesaplamak için destek fonksiyonları kullanılmış fakat bunun yaklaşım değerini iyileştirmede gözlemlenmiştir. Bununla beraber YBMG ile geliştirilen algoritmayı iyileştirmek için kullanılan yoğunluğa bağlı olarak pencereleri uyarlamalı hale getirme işlemi ÇYÇG algoritmasını da yüksek yoğunlukta iyileştirmiştir. Öncelikle YBMGMF9 algoritmasında yapıldığı gibi piksel değerlerinin simetrileri alınarak görüntü genişletilmiş ve bütün pikseller 7×7 pencere boyutuna da uygun hale getirilmiştir. Böylelikle gürültüsüz piksel 3×3 pencerede bulunmadığı

```

Test görüntüsünü oku;
Görüntünün sağına, soluna, yukarısına ve aşağısına üçer birimlik piksel
değerlerinin simetrikleri alınarak, görüntü boyutlarını artır;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;
Eğer bu pencere içerisinde 0 veya 255 dışında piksel değeri varsa
for Hedefteki pencere matrisi do
    Pencere içindeki gürültüsüz piksellerin ortalamasını hesapla;
    Pencere içindeki 0 ve 255 olan değerleri bu ortalama değer ile değiştir;
    Pencereye YBMG uygulayarak elde edilen tek değişkenli terimlerin
    medyanlarının ortalamasına sabit terimi ekleyerek merkez piksel yerine
    orijinal boyutlu görüntüde koy;
end
Değilse pencere boyutunu 5x5 olarak genişlet;
Eğer bu pencere içerisinde 0 veya 255 dışında piksel değeri varsa algoritmayı
uygula;
Değilse pencere boyutunu 7x7 olarak genişlet;
Eğer bu pencere içerisinde 0 veya 255 dışında piksel değeri varsa algoritmayı
uygula;
Elde edilen görüntüyü birer piksel genişleterek 3x3 boyutta işlemi üç kez
tekrar et.;
return İlk görüntü ile son görüntü arasındaki PSNR ve SSIM değerini döndür;

```

Algorithm 13: YBMGMF9 algoritması.

takdirde 5x5 pencerede, burada da bulunmadığı takdirde 7x7 pencerede ÇYÇG ile geliştirilen gürültü giderme algoritması çalıştırılmıştır. En son kalan gürültülü değerleri yakalayabilmek için birer birim genişletilmiş görüntüde 3x3 boyutta üç kez algoritma tekrar edilmiştir. Bu algorithmadan bu raporda ÇYÇGMF5 olarak bahsedilmiş, ÇYÇGMF5 algoritması Algoritma 14’de verilmiştir.

Yukarıda anlatılan algoritmalarından en iyi sonuç veren algoritma YBMGMF9 filtresinin son halinin detaylı anlatımı aşağıdaki gibidir ve bu filtre Yüksek Boyutlu Model Gösterilimi Medyan Filtresi, YBMGMF (High Dimensional Model Representation Median Filter, HDMRMF) olarak yayınlanmıştır [50].

- İlk olarak, boyutu $n_1 \times n_2$ olan görüntü $(n_1 + 6) \times (n_2 + 6)$ boyutuna büyütülür. Matrisi üçer birim genişletme işlemi, sınırdaki bulunan üç satır ve sütunun simetriklerinin alınmasıyla gerçekleştirilir. Algoritmada en fazla 7×7 boyutunda pencereler kullanıldığından, görüntüyü temsil eden matris her boyutta üç birim genişletilir. Bu, ilk satır ve sütundaki piksel değeri gürültülü olduğunda

```

Test görüntüsünü oku;
Görüntünün sağına, soluna, yukarısına ve aşağısına üçer birimlik piksel
değerlerinin simetrilerini kopyalayarak, görüntü boyutlarını genişlet;
Pikselleri birer kaydırarak gez, değeri 0 veya 255 olan pikselleri yakaladığın
yerde bu piksel merkezde olacak şekilde 3x3 pencere oluştur ;
Eğer bu pencere içerisinde 0 veya 255 dışında piksel değeri varsa
for Hedefteki pencere matrisi do
    Pencere içindeki gürültüsüz piksellerin ortalamasını hesapla;
    Pencere içindeki 0 ve 255 olan değerleri bu ortalama değer ile değiştir;
    Pencereye ÇYÇG uygulayarak elde edilen tek değişkenli terimlerin
    medyanlarının ortalamasına sabit terimi ekleyerek merkez piksel yerine
    orijinal görüntüde koy;
end
Değerle pencere boyutunu 5x5 olarak genişlet;
Eğer bu pencere içerisinde 0 veya 255 dışında piksel değeri varsa algoritmayı
uygula;
Değerle pencere boyutunu 7x7 olarak genişlet;
Eğer bu pencere içerisinde 0 veya 255 dışında piksel değeri varsa algoritmayı
uygula;
Elde edilen görüntüyü birer piksel genişleterek 3x3 boyutta işlemi üç kez
tekrar et.;
return İlk görüntü ile son görüntü arasındaki PSNR ve SSIM değerini döndür;

```

Algorithm 14: ÇYÇGMF5 algoritması.

ve uyarlanabilir pencere boyutu 7×7 olması gerektiğinde istenen koşulun sağlanmasını garanti eder.

- Boyutu $n_1 \times n_2$ olan gürültülü görüntüyü temsil eden matris, ilk satır ve sütundan başlayarak sistematik olarak incelenir. Değeri 0 veya 255 olan pikseller gürültülü pikseller olarak tanımlanır.
- Merkezinde gürültülü piksel değeri olan 3×3 boyutunda bir pencere oluşturulur ve bu pencere içinde gürültüsüz değerlerin varlığı kontrol edilir.
- Pencere içinde gürültüsüz değerler bulunursa, bulunan tüm gürültüsüz değerlerin ortalaması hesaplanır ve gürültülü pikselin yerine konulur.
- YBMG algoritması bu elde edilen matrise uygulanır. YBMG algoritmasından elde edilen 2.17 ifadesinde verilen sabit terim, 2.18 ifadesinde verilen tek değişkenli terimlerin medyanlarının ortalamasına eklenir ve bu değer, merkezdeki pikselin değeri olarak yerine konulur.

- Eğer 3×3 pencere içindeki tüm pikseller gürültülü ise, pencere boyutu 5×5 olarak genişletilir. 5×5 pencere içinde gürültüsüz piksellerin varlığı kontrol edilir. Bu boyutta gürültüsüz piksel değerleri bulunursa, yukarıda tanımlanan algoritma merkezdeki pikselin değerini belirlemek için uygulanır.
- Eğer 5×5 pencere içinde gürültüsüz piksel değerleri bulunamazsa, pencere boyutu son olarak 7×7 olarak genişletilir. Aynı algoritma 7×7 pencereye uygulanarak merkezdeki gürültülü piksel değeri hesaplanır.
- Görüntüdeki tüm piksel değerleri uyarlanabilir pencere boyutu kullanılarak incelendikten sonra, kalan gürültülü piksellerin olasılığını dikkate almak için algoritma 3×3 pencere kullanarak üç kez daha uygulanır. Bu şekilde, görüntüdeki tüm gürültülü piksel değerleri yakalanır.

YBMG ve ÇYÇG kullanılarak geliştirilen filtreler renkli görüntüler üzerine de uygulanmıştır. Renkli görüntülerde gürültü gidermek için burada geliştirilen algoritmalar, görüntü üç kanala ayrıldıktan sonra her bir kanala ayrı ayrı uygulanmıştır.

YBMG ve ÇYÇG ile oluşturulan algoritmaların görüntülere uygulanması ve elde edilen sonuçlar DeneySEL Sonuçlar bölümünde verilmiştir.



4. DENEYSEL BULGULAR

Bu bölümde, YBMG ve ÇYÇG ile oluşturulan filtrelerden elde edilen sonuçlar ayrıntılı olarak incelenmiştir. Oluşturulan algoritmaları analiz için üç temel metrik kullanılmıştır: Piksel Sinyal-Gürültü Oranı (PSNR), Yapısal Benzerlik İndeksi (SSIM) ve Çok Ölçekli Yapısal Benzerlik İndeksi (MS-SSIM).

PSNR, elde edilen görüntünün orijinal görüntüye ne kadar yakın olduğunu gösteren bir ölçüttür. Yüksek PSNR değeri, yeniden oluşturulan görüntünün orijinaline olan yakınlığının arttığını ifade eder. Diğer yandan, SSIM değeri 0 ile 1 arasında değişir ve bu değer arttıkça, görüntünün kalitesinin iyileştiği anlaşılır. SSIM, insan görsel algısının daha iyi taklit edilmesi amacıyla, parlaklık, kontrast ve yapı benzerliği gibi faktörleri dikkate alır [2].

Ayrıca, kenar detaylarının başarısını daha iyi ölçmek amacıyla MS-SSIM metriği kullanılmıştır. MS-SSIM, görüntünün farklı ölçeklerdeki yapısal benzerliğini değerlendirir ve değeri 1'e yaklaştıkça, görüntü kalitesinin ve kenar detaylarının daha başarılı olduğu anlaşılır. Bu metriklerin her biri, elde edilen görüntülerin orijinal görüntülerle ne derece uyumlu olduğunu ve görüntü kalitesini ne ölçüde koruduklarını belirlemek için kritik öneme sahiptir [51].

Şekil 4.1'de sunulan görüntülerde, orijinal görüntüye 0,01 varyanslı Gauss gürültüsü eklenmiştir. Bu gürültü eklendikten sonra, görüntüler sırasıyla ÇYÇGMF1, ÇYÇGMF2, ortalama ve medyan filtrelerle işlenerek gürültü giderilmiştir. Bu filtrelerin her biri, gürültünün azaltılması ve görüntü kalitesinin iyileştirilmesi amacıyla uygulanmıştır.

Benzer şekilde, Şekil 4.2'de gösterilen görüntülerde, orijinal görüntüye 0,05 varyanslı alacalı gürültü eklenmiştir. Bu durumda da, gürültü sırasıyla ÇYÇGMF1, ÇYÇGMF2, ortalama ve medyan filtrelerle giderilmiştir.

Kameraman, Barbara, Oturma Odası, Saat ve Kuş resimlerine 0,01 varyanslı Gauss gürültüsü eklenmiştir. Ardından, bu bozulmuş görüntüler sırasıyla ÇYÇGMF1,



(a)

(b)



(c)

(d)

(e)

(f)

Şekil 4.1 : (a) Orijinal görüntü, (b) 0,01 varyanslı Gauss gürültülü görüntü, (c) ÇYÇGMF1 filtresi, (d) ÇYÇGMF2 filtresi, (e) ortalama filtre, (f) medyan filtre

Çizelge 4.1 : 0,01 varyanslı Gauss gürültülü görüntülerin PSNR değerleri.

Görüntüler	ÇYÇGMF1	ÇYÇGMF2	Ortalama	Medyan
Kameraman	24,0541	25,4709	24,5281	25,0529
Barbara	22,7012	23,8211	23,0908	23,0491
Oturma odası	24,7368	25,5560	24,5914	24,2743
Saat	25,8862	27,1424	26,0571	25,6118
Kuş	24,2537	25,4054	24,4474	23,9530

ÇYÇGMF2, ortalama ve medyan filtreleri kullanılarak işlenmiş ve gürültü giderilmiştir. Her bir filtrenin uygulanması sonucunda elde edilen iyileştirilmiş görüntüler, Çizelge 4.1’de detaylı olarak sunulmuştur. Çizelgede sunulan sonuçlara bakıldığında, ÇYÇG’nin tek değişkenli terimleri ile geliştirilen ÇYÇGMF2 algoritmasının, ÇYÇG’nin sabit terimi ile oluşturulan algorithmadan ve temel filtreler olan ortalama ve medyan filtreden daha üstün performans sergilediği görülmektedir. ÇYÇGMF2 algoritması, gürültü giderme sürecinde daha yüksek doğruluk ve kalite sağlayarak, diğer filtrelere kıyasla daha iyi sonuçlar elde etmiştir.

Kameraman, Barbara, Oturma Odası, Saat ve Kuş resimlerine 0,05 varyanslı alacalı gürültü eklenmiştir. Sonrasında, bu bozulmuş görüntüler sırasıyla ÇYÇGMF1, ÇYÇGMF2, ortalama ve medyan filtreleri kullanılarak işlenmiş ve gürültü giderilmiştir. Her bir filtrenin uygulanması sonucu elde edilen düzeltilmiş görüntüler, Çizelge



(a)

(b)



(c)

(d)

(e)

(f)

Şekil 4.2 : (a)Orijinal görüntü, (b) 0,05 varyanslı alaca gürültülü görüntü, (c) ÇYÇGMF1 filtresi, (d) ÇYÇGMF2 filtresi, (e) ortalama filtre, (f) medyan filtre

Çizelge 4.2 : 0,05 varyanslı alacalı gürültülü görüntülerin PSNR değerleri.

Görüntüler	ÇYÇGMF1	ÇYÇGMF2	Ortalama	Medyan
Kameraman	24,1148	25,4573	24,1628	23,6412
Barbara	22,4330	23,4146	22,4584	21,6958
Oturma odası	24,6680	25,3933	24,2127	22,9620
Saat	26,6129	27,9490	26,5119	25,2032
Kuş	23,6661	24,6434	23,4875	22,5211

4.2’de detaylı olarak verilmiştir. Alacalı gürültüde de, gauss gürültüsünde olduğu gibi kıyaslama yapılan filtreler içinde ÇYÇGMF2 en iyi sonucu vermiştir.

YBMGMF1 olarak adlandırılan ve YBMG yöntemi kullanılarak tasarlanan birinci algoritma ile YBMG ile tasarlanan ikinci algoritma olan YBMGMF2 algoritması Çizelge 4.3’de tuz ve biber gürültüsünü gideren en temel filtre olan medyan filtre ile kıyaslanmıştır. 0,1 yoğunluktaki gürültüye sahip farklı görüntüye uygulanan algoritmalar neticesinde YBMGMF2 algoritması medyan filtreden daha iyi sonuçlar vermiştir.

Bu iki algoritma ve medyan filtre 0,1 yoğunluklu tuz ve biber gürültüsüne sahip Kameraman ve Monarch görüntülerine uygulandığında temizlenmiş görüntüler Şekil



(a)



(b)



(c)



(d)

Şekil 4.3 : (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF1 filtre, (c) YBMGMF2 filtre, (d) medyan filtre



(a)



(b)



(c)



(d)

Şekil 4.4 : (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF1 filtre, (c) YBMGMF2 filtre, (d) medyan filtre

4.3 ve Şekil 4.4’de verilmiş ve bunun sonucu olarak YBMGMF2 algoritmasının gürültüleri yok etmedeki başarısını görsel olarak da görmek oldukça mümkündür.

Çizelge 4.3 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMGMF1 ve YBMGMF2 ile elde edilen PSNR değerleri.

Görüntüler	Medyan filtre	YBMGMF1	YBMGMF2
Kameraman	27,0246	20,0001	31,7743
Barbara	24,4756	19,2674	31,0521
Oturma odası	26,7805	20,6823	32,2904
Monarch	27,0464	24,0480	33,0324
Kuş	28,6096	18,9470	33,3895

Çizelge 4.4 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMGMF3 ve YBMGMF4 ile elde edilen PSNR değerleri.

Görüntüler	Medyan filtre	YBMGMF3	YBMGMF4
Kameraman	27,1195	19,3945	32,3944
Barbara	24,4741	19,6459	31,7489
Oturma odası	26,5888	20,0935	33,4223
Monarch	27,2082	19,8948	33,7985
Kuş	28,6626	19,1240	34,7956



(a) (b) (c) (d)

Şekil 4.5 : (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF3 filtre, (c) YBMGMF4 filtre, (d) medyan filtre.



(a) (b) (c) (d)

Şekil 4.6 : (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF3 filtre, (c) YBMGMF4 filtre, (d) medyan filtre.

Sonuçları verilen algoritmaları geliştirebilmek için oluşturulan ve YBMGMF3 olarak adlandırılan filtre ile YBMGMF4 olarak adlandırılan filtre 0, 1 yoğunlukta tuz ve biber gürültüsü içeren Kameraman ve Monarch görüntülerine uygulandığında, temizlenmiş görüntüler Şekil 4.5 ve Şekil 4.6’de verilmiştir. Çizelge 4.4’de ise yine bu algoritmalar medyan filtre ile kıyaslama performanslarına yer verilmiştir. Çizelgede sunulan sonuçlara bakıldığında, YBMGMF4 algoritmasının, 0, 1 yoğunlukta gürültüye sahip görüntülerde, medyan filtreden çok daha yüksek PSNR değerlerine sahip olduğu görülmektedir. Bu durum, YBMGMF4 algoritmasının gürültü giderme konusunda medyan filtreye kıyasla daha üstün bir performans sergilediğini ve elde edilen görüntülerin orijinaline daha yakın olduğunu göstermektedir.

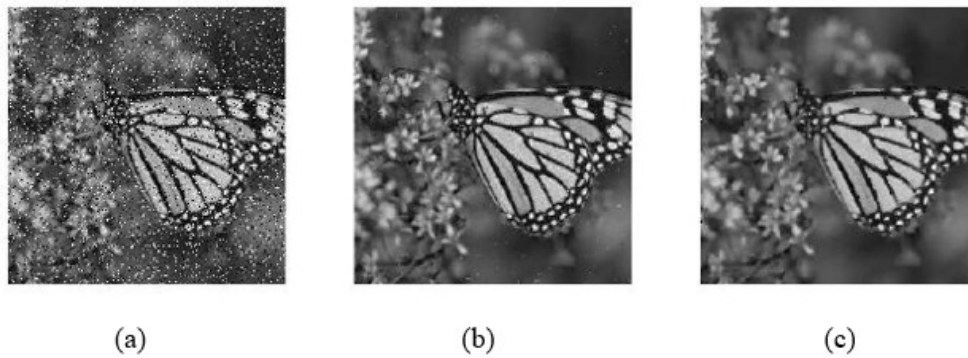
Çizelge 4.5 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMGMF5 ile elde edilen PSNR değerleri.

Görüntüler	Medyan filtre	YBMGMF5
Kameraman	27,0246	30,0883
Barbara	24,4756	30,2584
Oturma odası	26,7805	31,8046
Monarch	27,0464	30,7324
Kuş	28,6096	31,7921

Buraya kadar bahsedilen filtrelerden YBMGMF2 ve YBMGMF4 filtreleri medyan filtreden daha iyi sonuç vermekle beraber iterasyon gerektirmesi açısından işlem maliyeti daha yüksek algoritmalar. Bu işlem maliyetinden kurtulabilmek fakat aynı zamanda performansı düşürmemek adına tasarlanan algoritma YBMG5 algoritmasıdır. Bu filtre ile 0,1 yoğunluklu tuz ve biber gürültüsüne sahip farklı görüntüler temizlenerek elde edilen PSNR değerleri medyan filtre ile kıyaslanarak Çizelge 4.5’de verilmiştir. Sonuçlara bakıldığında YBMGMF2 ve YBMGMF4 algoritmalarının gerisinde fakat medyan filtreden daha iyi değerler bulunmuştur. YBMG5 algoritmasının 0,1 yoğunlukta tuz ve biber gürültüsüne sahip Kameraman ve Monarch görüntülerini temizlemiş görselleri Şekil 4.7 ve 4.8 ’de verilmiştir. Görsellerden de anlaşılacağı gibi YBMG5 filtresi düşük yoğunlukta gürültüleri gidermekte oldukça başarılıdır.



Şekil 4.7 : (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF5 filtre, (c) medyan filtre.



Şekil 4.8 : (a) 0,1 yoğunluklu tuz ve biber gürültülü görüntü, (b) YBMGMF5 filtre, (c) medyan filtre.

Çizelge 4.6’de buraya kadar YBMG ile tasarlanan YBMGMF1, YBMGMF2, YBMGMF3, YBMGMF4 ve YBMGMF5 algoritmalar birbirleri ile kıyaslanmıştır. PSNR değerleri göstermektedir ki, iterasyon gerektiren YBMGMF4 algoritması 0,1

Çizelge 4.6 : 0,1 yoğunluklu tuz ve biber gürültüsüne sahip siyah beyaz görüntülerin YBMG algoritmaları ile elde edilen PSNR değerleri.

Görüntüler	YBMGMF1	YBMGMF2	YBMGMF3	YBMGMF4	YBMGMF5
Kameraman	20,0001	31,7743	19,3945	32,3944	30,0883
Barbara	19,2674	31,0528	19,6459	31,7489	30,2584
Oturma odası	20,6823	32,2904	20,0935	33,4223	31,8046
Monarch	24,0480	33,0324	19,8948	33,7985	30,7324
Kuş	18,9470	33,3895	19,1240	34,7956	31,7921

yoğunluktaki tuz ve biber gürültüsünü gidermekte diğerlerinden daha başarılıdır. Bunu takiben YBMGMF2 ve YBMGMF5 algoritmaları gelmekle beraber YBMGMF1 ve YBMGMF3 diğerlerinde göre oldukça geride kalmaktadır. YBMGMF1 ve YBMGMF3 algoritmalarının YBMGMF2 ve YBMGMF4'den farkının tekrarlamalı olmadığı dikkate alınır, bu algoritmalar tek seferde oldukça az sayıda gürültülü pikselleri yakalayabilmektedir. Buna karşın YBMG5 algoritması tek seferde gürültülü pikselleri tespit etmekle beraber iteratif olan yöntemler kadar yüksek başarılarla ulaşamamıştır.

Çizelge 4.7 : 0,1, 0,2 ve 0,3 yoğunluklu tuz ve biber gürültüsüne sahip Monarch görüntüsünün farklı filtrelerden elde edilen PSNR değerleri.

Yoğunluk	MF	FUMF	KUIA	PYTF	YBMGMF2	YBMGMF4	YBMGMF5
0,1	27,046	37,858	24,853	34,515	33,032	33,798	30,732
0,2	24,225	33,381	22,138	30,162	27,527	28,254	25,273
0,3	22,062	31,064	19,969	30,162	24,040	24,808	22,029

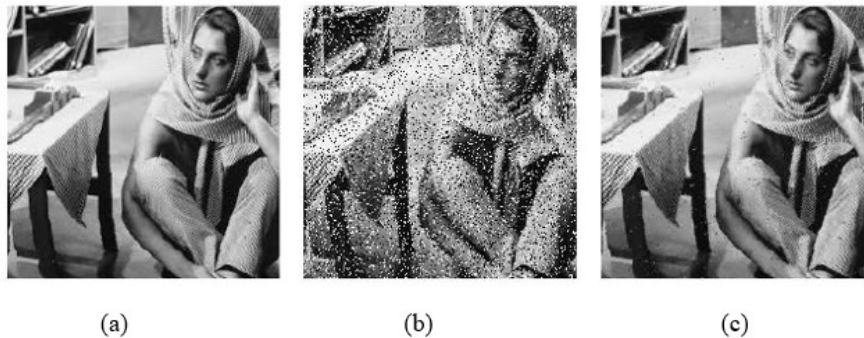
Çizelge 4.7'da sunulan tablo, YBMG kullanılarak tuz ve biber gürültüsünü gidermek amacıyla geliştirilen filtrelerin verimliliğini, medyan filtre ve literatüre kazandırılmış yeni çalışmalarla karşılaştırmaktadır. Bu tabloda medyan filtre MF ile gösterilmiştir, FUMF olarak kısaltılan algoritma Farklı Uygulanabilen Medyan Filtre (Different Applied Median Filtre, DAMF) [2], KUIA olarak kısaltılan algoritma Küresel Uyarlanabilir Kesirli İntegral Algoritması (Global Adaptive Fractional Integral Algorithm, GAFIA) [19], PYTF olarak kısaltılan algoritma Piksel Yoğunluğu Tabanlı Filtre (Based on Pixel Density Filter, BPDF) [20] olarak ifade edilmiş ve bundan sonra yer alan tablolarda da aynı gösterimleri mevcuttur. Elde edilen sonuçlar, bu yeni algoritmaların, kıyaslama yapılan 4 farklı filtrenin 2 tanesinden daha iyi performans gösterdiğini ortaya koymaktadır. Geliştirilen bu algoritmalarda, gürültü yoğunluğu arttıkça filtre verimliliğinin düştüğü gözlemlenmiştir. Bu durum çalışmaları yoğunluk

yükseldikçe de performansın düşmeyeceği araştırmaya yapmaya yöneltmiştir. Bu arayış sonucunda geliştirilen ve bundan sonraki anlatılacak olan filtrelerde bu problemin üstesinden gelinmiş ve hem düşük hem yüksek yoğunlukta başarılar elde edilemeye çalışılmıştır.

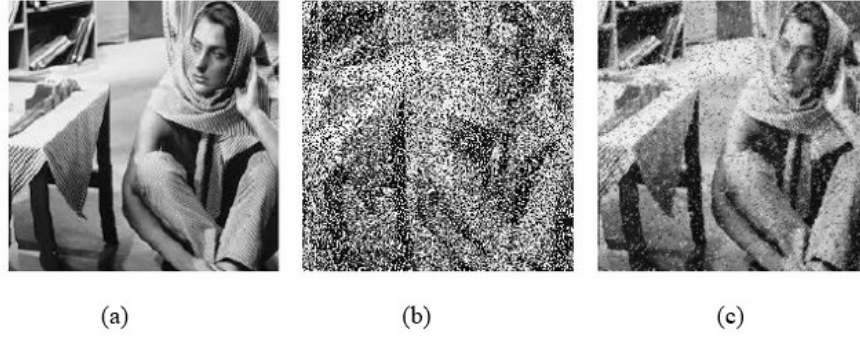
Hibrit bir model olan YBMGMF6 yöntemi ile sonuçları iyileştirmek hedeflenmekle beraber elde edilen sonuçlar literatürün iyi sonuç veren yöntemleriyle karşılaştırıldığında istenildiği kadar yüksek bir iyileştirme oranı elde edilememiştir. Sonuçlar Çizelge 4.8’de verilmiştir. Güncel çalışmalarla yapılan kıyaslamada YBMGMF6 yöntemi geride kalmıştır fakat Şekil 4.9’de verile 0,1 tuz ve biber gürültüsüne sahip Barbara görüntüsüne bakıldığında görsel olarak yine de gürültüleri gidermede yöntemin başarılı olduğu söylenebilir. Şekil 4.10’da verilen 0,5 yoğunluklu görüntüye bakıldığında ise gürültülerin daha az miktarda arındırıldığı gözlemlenmektedir.

Çizelge 4.8 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF6
Lena	PSNR	39,1514	35,8617	38,2292	31,9468
	SSIM	0,9786	0,9654	0,9767	0,8821
Kameraman	PSNR	39,0328	34,8175	38,1744	32,9228
	SSIM	0,9906	0,9792	0,9882	0,8797
Oturma Odası	PSNR	34,8566	32,1853	34,3573	31,8313
	SSIM	0,9651	0,9434	0,9625	0,9031
Barbara	PSNR	30,4355	29,2320	30,8492	29,2341
	SSIM	0,9590	0,9424	0,9599	0,8918
Monarch	PSNR	33,3812	30,1621	32,2762	31,1282
	SSIM	0,9823	0,9653	0,9758	0,9143



Şekil 4.9 : (a) Orijinal görüntü, (b) 0,1 gürültülü görüntü, (c) YBMGMF6 ile temizlenmiş görüntü.



Şekil 4.10 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF6 ile temizlenmiş görüntü.

Yüksek yoğunlukta daha yüksek başarılarla ulaşabilmek için pencere boyutunun daha büyük olması gerektiği varsayımıyla YBMGMF7 yöntemi tasarlanmıştır. Bu sayede hem düşük yoğunlukta hem de yüksek yoğunlukta literatürdeki iyi çalışmalarla kıyaslanabilir sonuçlar elde edilmiştir. YBMGMF7 yöntemi, iteratif olmadığı için oldukça hızlı çalışan ve etkili bir yöntem olarak öne çıkmaktadır.

YBMGMF7 algoritmasından elde edilen sonuçlar, literatürün güncel yöntemlerinden FUMF, PYTF ve UTTKA ile karşılaştırılarak Çizelge 4.9'de sunulmuştur. Sonuçlar göstermektedir ki, 0,2 yoğunluklu gürültüye sahip görüntülerde, YBMGMF7 algoritması PSNR değeri açısından 5 farklı görüntünün 4'ünde en iyi sonucu vermiştir. Ayrıca, SSIM değeri açısından yapılan karşılaştırmalarda ise tüm görüntülerde en iyi sonuca ulaşmıştır.

YBMGMF7 algoritması kullanılarak gürültüden arındırılan görüntüler, Şekil 4.11 ve Şekil 4.12'de sunulmuştur. Bu görsellerde, 0,2 yoğunlukta tuz ve biber gürültüsüne sahip Kameraman görüntüsü ve 0,5 yoğunlukta gürültü içeren Barbara görüntüsü kullanılmıştır. Her iki durumda da, gürültülerin etkili bir şekilde azaltıldığı ve görüntülerin temizlendiği görülmektedir. Özellikle 0,5 yoğunlukta gürültü içeren Barbara görüntüsünde elde edilen temizlenme başarısı, YBMGMF7 algoritmasının yüksek gürültü yoğunluklarına karşı da etkili olduğunu göstermektedir.

YBMGMF8 algoritması, daha önce geliştirilmiş olan YBMGMF6 ve YBMGMF7 algoritmalarının birleştirilmesi sonucu oluşturulmuş ve hibrit model büyütilen pencereye uygulayan bir modeldir. Bu algoritmanın tasarımı, iki önceki modelin güçlü yönlerini bir araya getirerek daha etkili bir performans elde etmeyi amaçlamaktadır. Yapılan bu birleştirme işleminin deneysel sonuçlara nasıl yansıdığı Çizelge 4.10'de

Çizelge 4.9 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF7
Lena	PSNR	39,1514	35,8617	38,2292	37,0582
	SSIM	0,9786	0,9654	0,9767	0,9796
Kameraman	PSNR	39,0328	34,8175	38,1744	40,5259
	SSIM	0,9906	0,9792	0,9882	0,9918
Oturma Odası	PSNR	34,8566	32,1853	34,3573	35,5647
	SSIM	0,9651	0,9434	0,9625	0,9695
Barbara	PSNR	30,4355	29,2320	30,8492	31,1346
	SSIM	0,9590	0,9424	0,9599	0,9628
Monarch	PSNR	33,3812	30,1621	32,2762	33,9114
	SSIM	0,9823	0,9653	0,9758	0,9831

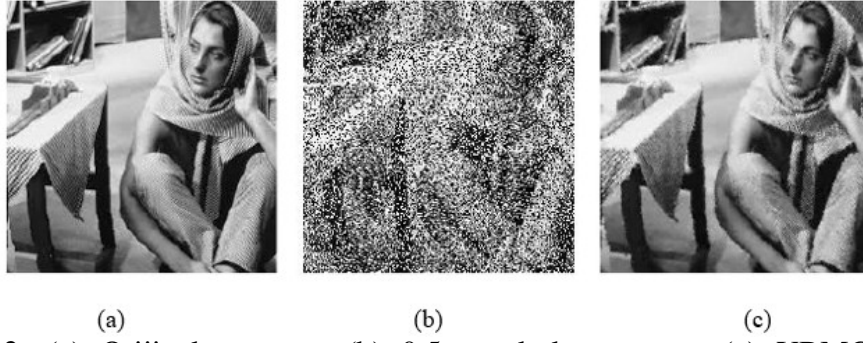


Şekil 4.11 : (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.

detaylı bir şekilde gösterilmiştir. Çizelgedeki sonuçlar incelendiğinde, YBMGMF7 ile benzer sonuçların elde edilmesi, YBMG modelinin tek başına da yüksek bir başarıya ulaşabildiğini göstermektedir.

Bu yeni algoritma, 0,2 yoğunlukta tuz ve biber gürültüsüne sahip olan Kameraman görüntüsü ve 0,5 yoğunlukta tuz ve biber gürültüsüne sahip olan Barbara görüntülerini gürültülerden arındırmak için uygulanmıştır. Bu işlem sonrasında elde edilen görseller, Şekil 4.13 ve Şekil 4.14’de sunulmuştur. Görsellerin incelenmesi sonucunda, YBMGMF7 algoritması ile benzer bir temizlik oranının elde edildiği gözlemlenmiştir. Bu da, YBMGMF8 algoritmasının, önceki modellerin etkinliğini koruyarak, başarılı bir şekilde gürültü giderme işlemi gerçekleştirdiğini göstermektedir.

Çizelge 4.11’de, farklı yoğunluklarda tuz ve biber gürültüsüne sahip Monarch görüntüsünün, FUMF, PYTF, UTTKA, YBMGMF7 ve YBMGMF8 filtreleri kullanılarak temizlenmesi sonucu elde edilen veriler sunulmuştur. Bu çizelge, her bir filtrenin farklı yoğunluklardaki gürültülerle başa çıkma yeteneğini karşılaştırmalı



Şekil 4.12 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.

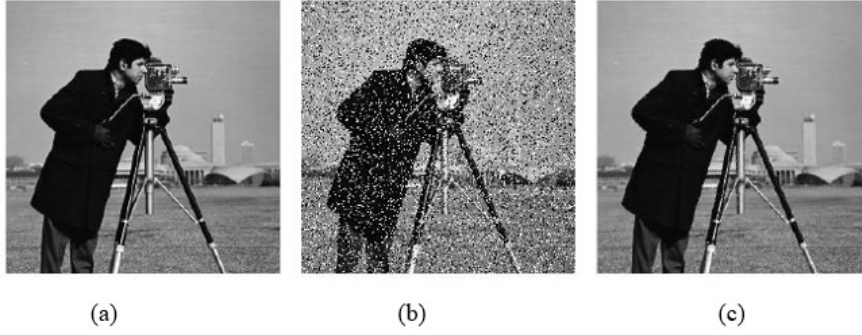
Çizelge 4.10 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF8
Lena	PSNR	39,1514	35,8617	38,2292	37,1459
	SSIM	0,9786	0,9654	0,9767	0,9795
Kameraman	PSNR	39,0328	34,8175	38,1744	40,5108
	SSIM	0,9906	0,9792	0,9882	0,9916
Oturma Odası	PSNR	34,8566	32,1853	34,3573	35,4726
	SSIM	0,9651	0,9434	0,9625	0,9689
Barbara	PSNR	30,4355	29,2320	30,8492	31,2217
	SSIM	0,9590	0,9424	0,9599	0,9633
Monarch	PSNR	33,3812	30,1621	32,2762	33,8641
	SSIM	0,9823	0,9653	0,9758	0,9831

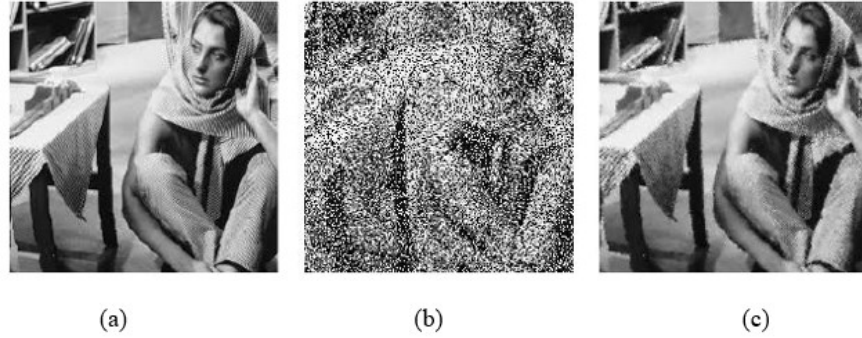
Çizelge 4.11 : Monarch görüntüsünün farklı yoğunluklardaki farklı filtrelerle elde edilmiş PSNR ve SSIM değerleri.

Yoğunluk	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF7	YBMGMF8
0.3	PSNR	31,0643	27,2616	30,0231	31,6490	31,7159
	SSIM	0,9710	0,9366	0,9597	0,9719	0,9717
0.4	PSNR	29,0522	24,6550	28,2791	29,4862	29,5337
	SSIM	0,9547	0,8945	0,9382	0,9557	0,9559
0.5	PSNR	27,4164	22,5105	26,9119	27,8392	27,5013
	SSIM	0,9344	0,8399	0,9147	0,9348	0,9329
0.6	PSNR	25,7691	19,9462	25,3341	25,7671	25,6111
	SSIM	0,9072	0,7511	0,8811	0,9029	0,9011
0.7	PSNR	24,2804	17,4889	23,9385	23,4128	22,3071
	SSIM	0,8721	0,6360	0,8376	0,8478	0,8485
0.8	PSNR	22,5909	14,2249	22,0710	20,8103	20,2703
	SSIM	0,8199	0,4495	0,7651	0,7522	0,7592

olarak göstermektedir. Elde edilen sonuçlar, 0.6 yoğunluğun altındaki gürültülerde, YBMGMF7 ve YBMGMF8 filtrelerinin diğer yöntemlerden daha iyi performans sergilediğini ortaya koymuştur.



Şekil 4.13 : (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) YBMGMF8 ile temizlenmiş görüntü.



Şekil 4.14 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF8 ile temizlenmiş görüntü.

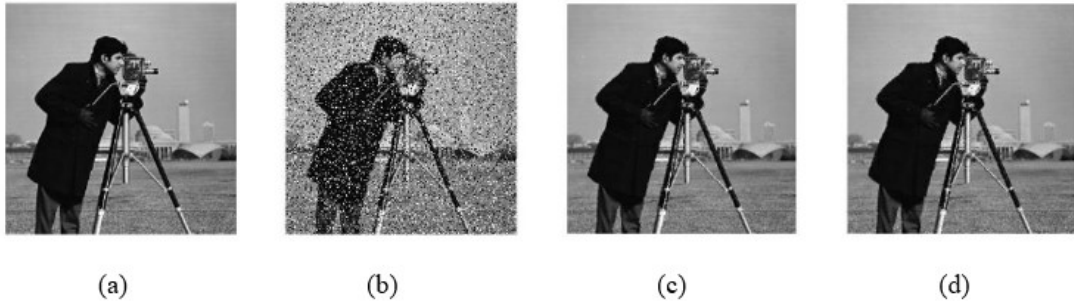
Ancak, tuz ve biber gürültüsünün yoğunluğu arttıkça, YBMGMF7 ve YBMGMF8 filtrelerinin performansında bir düşüş gözlemlenmiştir. Bu yüksek yoğunluklu gürültü durumlarında, her ne kadar YBMGMF7 ve YBMGMF8 filtreleri hala iyi sonuçlar vermeye devam etse de, diğer filtrelerin en iyisi olma özelliğini kaybetmişlerdir. Bu durum, YBMGMF7 ve YBMGMF8 yöntemlerinin düşük ve orta yoğunluktaki gürültülerde üstün performans sergilediğini, ancak yüksek yoğunluktaki gürültülerde diğer yöntemlerle kıyaslandığında daha az etkili olduğunu göstermektedir.

ÇYÇG ile üretilen ve çalışma mekanizması YBMGMF7 ile benzerlik gösteren ÇYÇGMF3 algoritması ve ÇYÇGMF4 olarak adlandırılan YBMGMF7 ile ön filtrelemenin yapıldığı algoritmaya ait sonuçlar, literatürün güncel çalışmaları ile kıyaslanarak Çizelge 4.12’de verilmiştir. 0,2 yoğunluk düzeyinde filtrelenen görüntülerden elde edilen PSNR değerleri iyi olmakla beraber tüm yöntemlerin en iyisi değildir. Çizelge 4.12’de elde edilen değerler göre ÇYÇGMF3 ve ÇYÇGMF4 filtreleri birbirleri ile çok yakın sonuçlara sahiptir ve incelenen 5 görüntü içerisinde Barbara görüntüsünde ÇYÇGMF3 filtresi diğer yöntemlerden daha iyi çalışmıştır.

ÇYÇGMF3 ve ÇYÇGMF4 algoritmaları ile gürültüden arındırılan görüntüler Şekil 4.15 ve Şekil 4.16’da verilmiştir. Burada 0,2 ve 0,5 yoğunluklu tuz ve biber gürültüsüne sahip Kameraman görüntülerine denenen filtreler görsel olarak gürültüleri temizlemekte oldukça başarılıdır.

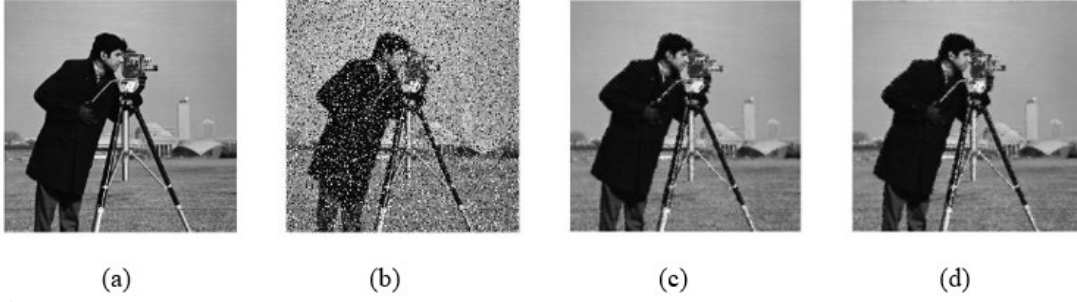
Çizelge 4.12 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerden elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	FUMF	PYTF	UTTKA	ÇYÇGMF3	ÇYÇGMF4
Lena	PSNR	39,1514	35,8617	38,2292	35,9807	35,5943
	SSIM	0,9786	0,9654	0,9767	0,9767	0,9736
Kameraman	PSNR	39,0328	34,8175	38,1744	38,7313	38,4059
	SSIM	0,9906	0,9792	0,9882	0,9897	0,9866
Oturma Odası	PSNR	34,8566	32,1853	34,3573	34,7943	34,4081
	SSIM	0,9651	0,9434	0,9625	0,9656	0,9623
Barbara	PSNR	30,4355	29,2320	30,8492	30,9307	30,9210
	SSIM	0,9590	0,9424	0,9599	0,9612	0,9527
Monarch	PSNR	33,3812	30,1621	32,2762	32,8349	32,4801
	SSIM	0,9823	0,9653	0,9758	0,9800	0,9740



Şekil 4.15 : (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) ÇYÇGMF3 ile temizlenmiş görüntü, (d) ÇYÇGMF4 ile temizlenmiş görüntü.

YBMG ve ÇYÇG ile elde edilen YBMGMF6, YBMGMF7, YBMGMF8, ÇYÇGMF3 ve ÇYÇGMF4 filtrelerinin sonuçları kendi içinde kıyaslanmıştır. Çizelge 4.13 elde edilen sonuçlar göstermektedir ki 0,2 yoğunlukta tuz ve biber gürültüsüne sahip farklı görüntülerde YBMGMF7 ve YBMGMF8 algoritmaları ÇYÇGMF3 ve ÇYÇGMF4 algoritmalarından daha iyi sonuç vermiştir. YBMGMF7, YBMGMF8, ÇYÇGMF3 ve ÇYÇGMF4 görüntüleri farklı yoğunluk değerlerine sahip Kameraman görüntüsüne uygulanmış ve elde edilen sonuçlar Çizelge 4.14’de verilmiştir. Bu sonuçlara göre gürültü yoğunluğu arttıkça ÇYÇG ile tasarlanan algoritmalar yüksek yoğunluk düzeylerinde YBMG ile elde edilen algoritmalarından daha başarılıdır.

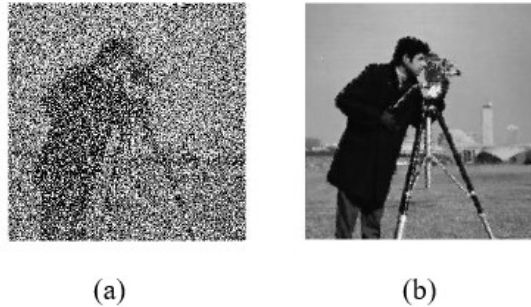


Şekil 4.16 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) ÇYÇGMF3 ile temizlenmiş görüntü, (d) ÇYÇGMF4 ile temizlenmiş görüntü.

0,6, 0,7, 0,8 ve 0,9 yoğunluktaki tuz ve biber gürültüsüne sahip Kameraman görüntüsünü YBMGMF7 algoritması ile temizlenerek görüntüler Şekil 4.17, Şekil 4.18, Şekil 4.19 ve Şekil 4.20’de verilmiştir. YBMGMF7 algoritmasının renk geçişlerinin fazla olduğu görüntülerde bu geçişleri koruduğu gözlemlenmiştir. Bununla ilgili örnekler Şekil 4.21, 4.22 ve 4.23’de verilmiştir.

Çizelge 4.13 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin YBMG ve ÇYCG filtreleri ile elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	YBMGMF6	YBMGMF7	YBMGMF8	ÇYÇGMF3	ÇYÇGMF4
Lena	PSNR	31,9468	37,0582	37,1459	35,9807	35,5943
	SSIM	0,8821	0,9796	0,9795	0,9897	0,9736
Kameraman	PSNR	32,9228	40,5259	40,5108	38,7313	38,4059
	SSIM	0,8797	0,9918	0,9916	0,9897	0,9866
Oturma Odası	PSNR	31,8313	35,5647	35,4726	34,7943	34,4081
	SSIM	0,9031	0,9695	0,9689	0,9656	0,9623
Barbara	PSNR	29,2341	31,1346	31,2217	30,9307	30,9210
	SSIM	0,8918	0,9628	0,9633	0,9552	0,9591
Monarch	PSNR	31,1282	33,9114	33,8641	32,8349	32,4801
	SSIM	0,9143	0,9831	0,9831	0,9800	0,9740

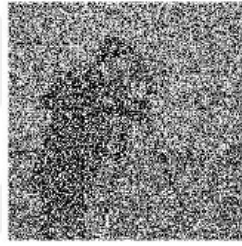


Şekil 4.17 : (a)Orijinal görüntü, (b) 0,6 gürültülü görüntü, YBMGMF7 ile temizlenmiş görüntü.

YBMG algoritmaları arasında, dinamik pencere yapısı kullanılarak yüksek yoğunluk-taki gürültü oranlarını gidermeyi amaçlayan YBMGMF9 algoritması geliştirilmiştir.

Çizelge 4.14 : Kameraman görüntüsünün farklı yoğunluklardaki CYCG ve YBMG filtrelerinden elde edilen PSNR ve SSIM değerleri.

Yoğunluk	Ölçüler	YBMGMF7	YBMGMF8	ÇYÇGMF3	ÇYÇGMF4
0,3	PSNR	37,8656	37,8546	36,2986	36,0158
	SSIM	0,9852	0,9851	0,9825	0,9773
0,4	PSNR	35,7752	35,8093	34,6122	34,0321
	SSIM	0,9768	0,9764	0,9736	0,9662
0,5	PSNR	33,2133	33,2430	32,8332	32,3029
	SSIM	0,9636	0,9631	0,9609	0,9522
0,6	PSNR	31,1593	31,0928	30,5146	30,7074
	SSIM	0,9438	0,9434	0,9399	0,9346
0,7	PSNR	28,6646	28,8336	28,6007	28,5633
	SSIM	0,9121	0,9130	0,9111	0,9042
0,8	PSNR	25,8916	25,6423	25,8278	25,9428
	SSIM	0,8587	0,8572	0,8596	0,8536
0,9	PSNR	22,0699	22,4825	22,4400	22,4632
	SSIM	0,7645	0,7686	0,7745	0,7636



(a)



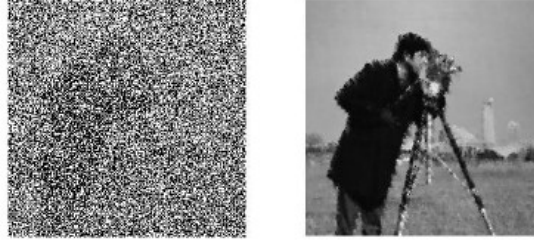
(b)

Şekil 4.18 : (a) Orijinal görüntü, (b) 0,7 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.

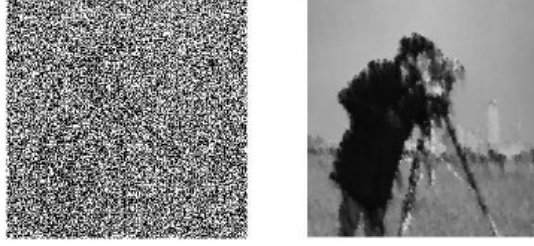
Bu algoritma ile elde edilen sonuçlar, Çizelge 4.15’de diğer filtrelerle karşılaştırmalı olarak sunulmuştur. Sonuçlar incelendiğinde, YBMGMF9 algoritmasının 0,2 yoğunlukta, karşılaştırılan tüm filtrelerden daha iyi performans gösterdiği görülmüştür.

YBMGMF9 algoritması ile gürültülerinden arındırılan görüntüler ise Şekil 4.24, Şekil 4.25 ve Şekil 4.26’te sunulmuştur. Bu görsellerin incelenmesi, hem düşük hem de oldukça yüksek yoğunluklardaki gürültülerde çok iyi görüntü kalitesinin elde edildiğini göstermektedir. YBMGMF9 algoritması, yüksek yoğunlukta gürültü ortamlarında bile etkili bir şekilde çalışarak, temiz ve net görüntüler elde edilmesini sağlamıştır.

YBMGMF9 algoritmasının çalışma prensibi ÇYÇG’ye uyarlanarak geliştirilen ÇYÇGMF5 algoritmasından elde edilen sonuçlar, Çizelge 4.16’de sunulmuştur.



Şekil 4.19 : (a) Orijinal görüntü, (b) 0,8 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.



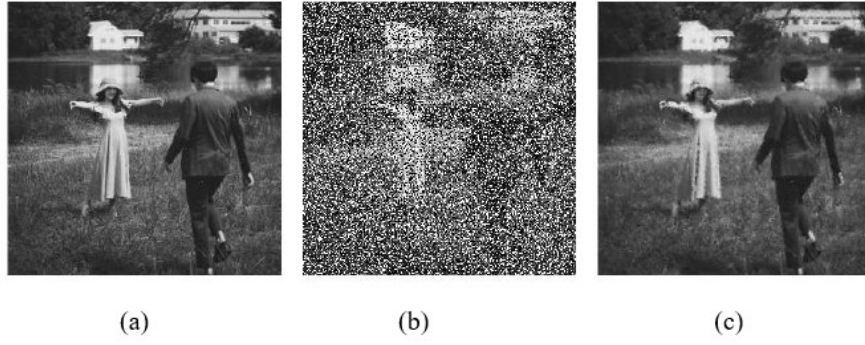
Şekil 4.20 : (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.

Çizelge 4.15 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.

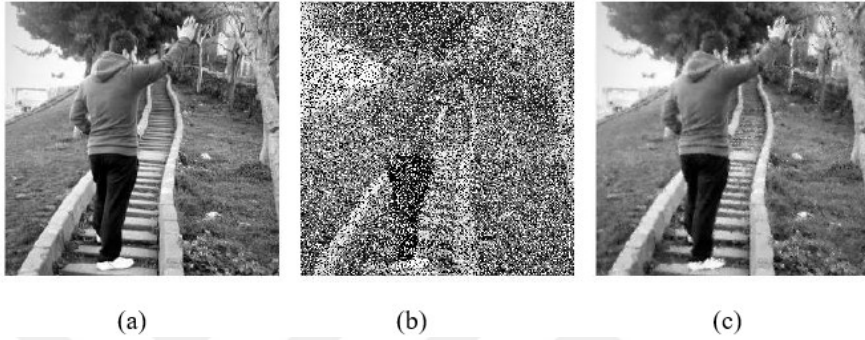
Görüntüler	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF9
Ev	PSNR	36,5087	34,0610	36,2168	37,1344
	SSIM	0,9650	0,9467	0,9645	0,9684
Kameraman	PSNR	39,2685	34,8175	38,1744	40,7923
	SSIM	0,9899	0,9792	0,9882	0,9919
Oturma Odası	PSNR	35,1134	32,1853	34,3573	35,6109
	SSIM	0,9663	0,9434	0,9625	0,9690
Barbara	PSNR	30,4267	29,2320	30,8492	31,2374
	SSIM	0,9585	0,9424	0,9599	0,9636
Monarch	PSNR	33,6317	30,1621	32,2762	34,1784
	SSIM	0,9820	0,9653	0,9758	0,9833

Bu sonuçlar, kıyaslanan filtreler arasında en iyi performansın YBMGMF9 algoritmasından, ikinci en iyi performansın ise ÇYÇGMF5 algoritmasından elde edildiğini göstermektedir. Bu bulgular, ÇYÇG ve dinamik pencere yapısının bir araya getirildiğinde, tuz ve biber gürültüsünü gidermede oldukça başarılı sonuçlar doğurduğunu kanıtlamaktadır.

Elde edilen bu veriler, ÇYÇGMF5 algoritmasının, YBMGMF9 algoritmasının güçlü yönlerini taşıyarak yüksek performans gösterdiğini ve bu iki yaklaşımın yüksek



Şekil 4.21 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.



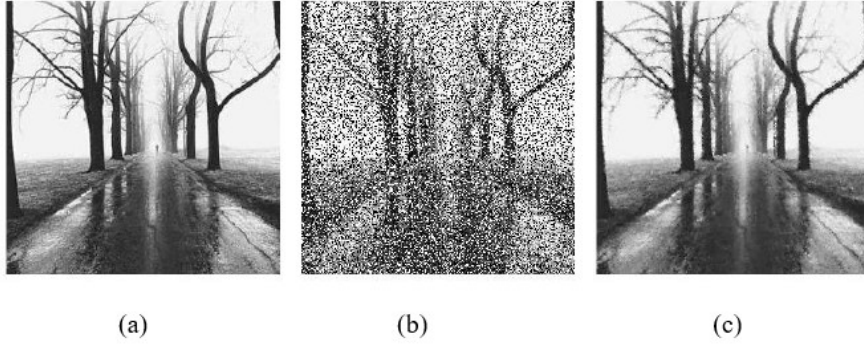
Şekil 4.22 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.

yoğunluklu gürültü oranlarında da etkin bir gürültü giderme sağladığını ortaya koymaktadır.

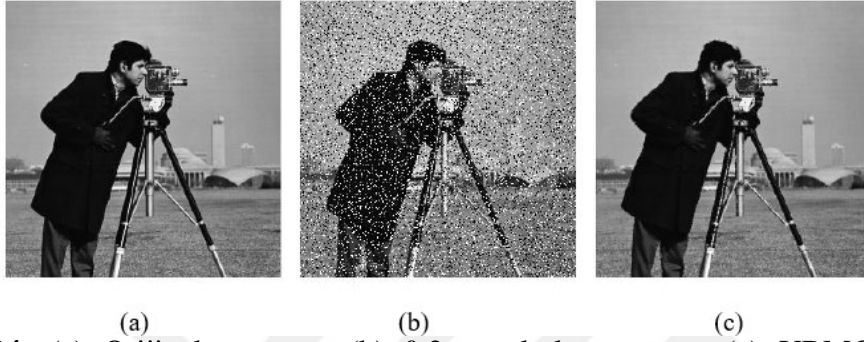
ÇYÇGMF5 algoritması ile gürültülerinden arındırılan farklı yoğunlukta Kamera man görüntüleri Şekil 4.27, 4.28 ve 4.29’de gösterilmiştir. Bu görsellerde de ÇYÇGMF5 algoritmasının gürültüleri gidermekte oldukça başarılı olduğu görülmektedir.

Çizelge 4.16 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin farklı filtrelerle elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF9	ÇYÇGMF5
Ev	PSNR	36,5087	34,0610	36,2168	37,1344	36,6784
	SSIM	0,9650	0,9467	0,9645	0,9684	0,9669
Kameraman	PSNR	39,2685	34,8175	38,1744	40,7923	39,2788
	SSIM	0,9899	0,9792	0,9882	0,9919	0,9905
Oturma Odası	PSNR	35,1134	32,1853	34,3573	35,6109	34,9209
	SSIM	0,9663	0,9434	0,9625	0,9690	0,9661
Barbara	PSNR	30,4267	29,2320	30,8492	31,2374	31,0534
	SSIM	0,9585	0,9424	0,9599	0,9636	0,9622
Monarch	PSNR	33,6317	30,1621	32,2762	34,1784	32,8136
	SSIM	0,9820	0,9653	0,9758	0,9833	0,9794



Şekil 4.23 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF7 ile temizlenmiş görüntü.

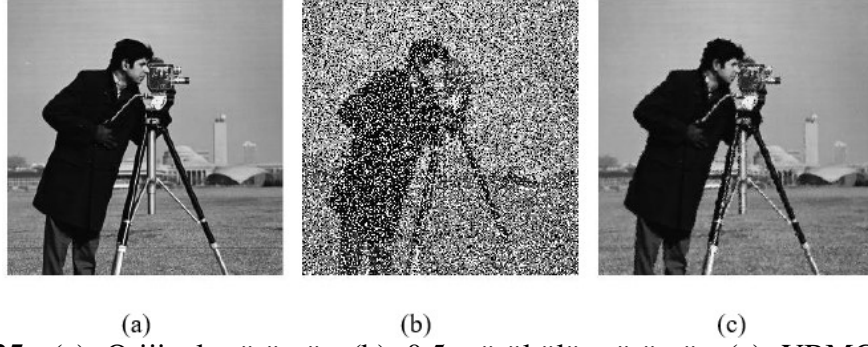


Şekil 4.24 : (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) YBMGMF9 ile temizlenmiş görüntü.

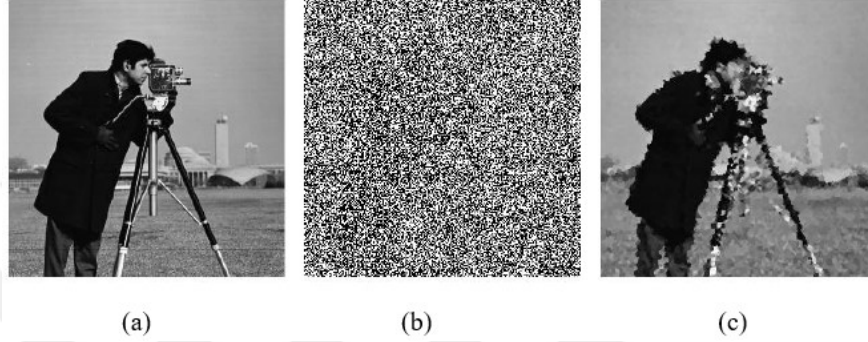
Çizelge 4.17 : Monarch görüntüsünün farklı yoğunluklardaki farklı filtrelerle elde edilen PSNR ve SSIM değerleri.

Yoğunluk	Ölçüler	FUMF	PYTF	UTTKA	YBMGMF9	ÇYÇGMF5
0,3	PSNR	31,0643	27,2616	30,0231	31,7928	30,8036
	SSIM	0,9710	0,9366	0,9597	0,9726	0,9681
0,4	PSNR	29,0522	24,6550	28,2791	29,8066	29,4119
	SSIM	0,9547	0,8945	0,9382	0,9574	0,9548
0,5	PSNR	27,4164	22,5105	28,1665	27,8392	27,9614
	SSIM	0,9344	0,8399	0,9147	0,9399	0,9368
0,6	PSNR	25,7691	19,9462	25,3341	26,2717	26,0756
	SSIM	0,9072	0,7511	0,8811	0,9113	0,9089
0,7	PSNR	24,2804	17,4889	23,9385	24,5984	24,6275
	SSIM	0,8721	0,6360	0,8376	0,8757	0,8760
0,8	PSNR	22,5909	14,2249	22,0710	22,7934	22,8172
	SSIM	0,8199	0,4495	0,7651	0,8213	0,8222
0,9	PSNR	19,9265	10,1408	19,0473	20,1052	20,2917
	SSIM	0,7085	0,2635	0,6793	0,7163	0,7216

Çizelge 4.17’de farklı yoğunluk derecelerinde tuz ve biber gürültüsüne sahip Monarch görüntüsünün FUMF, PYTF, UTTKA, YBMGMF9 ve ÇYÇGMF5 filtreleri ile elde edilen PSNR ve SSIM değerleri verilmiştir. Elde edilen sonuçlar göstermektedir ki YBMGMF9 ve ÇYÇGMF5 yoğunluk derecesi yükseldikçe ÇYÇGMF5 algoritması



Şekil 4.25 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) YBMGMF9 ile temizlenmiş görüntü.

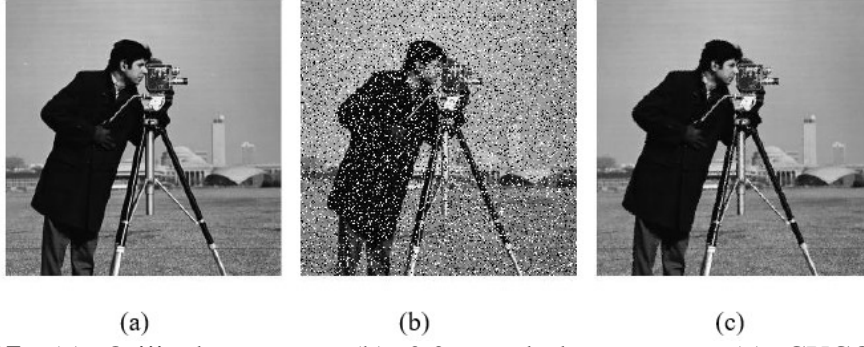


Şekil 4.26 : (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) YBMGMF9 ile temizlenmiş görüntü.

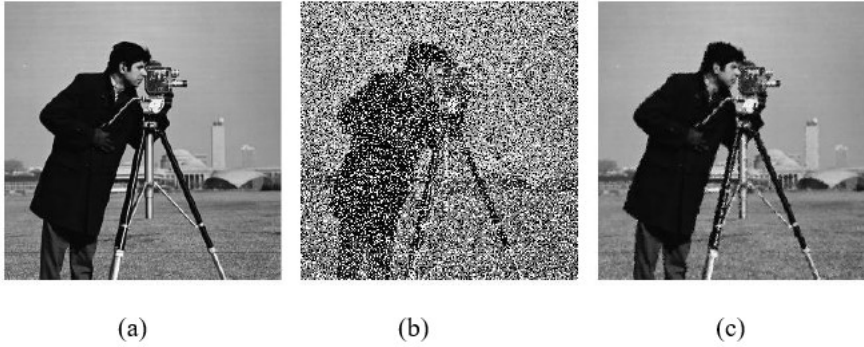
YBMGMF9 algoritmasından daha iyi sonuçlar vermiştir ve diğer yöntemlerden üstün başarı sağlamıştır.

İteratif Ağırlıklı Ortalama Filtre, IAOF [32] literatür bölümünde sonuçları iyi olan güncel çalışmalardan biri olarak verilmiştir fakat kıyaslama yapılabilecek görüntü olarak çalışmada Lena resmi kullanılmıştır. Bu çalışmada görüntü olarak Lena'ya yer verilmediği için kıyaslama tablosu bulunmamaktadır. Lena görüntüsü için 0,1 yoğunluklu tuz ve biber gürültüsü için FUMF [2] PSNR değeri 43,2, IAOF [32] PSNR değeri 43,3, 0,5 yoğunluklu tuz ve biber gürültüsü için FUMF [2] PSNR değeri 33,1 IAOF [32] PSNR değeri 34,0, 0,9 yoğunluklu tuz ve biber gürültüsü için FUMF [2] PSNR değeri 27,0 IAOF [32] PSNR değeri 27,1 olarak bulunmuştur. Bu sonuçlarda bizim çalışmamızın kıyaslanabilir iyilikte olduğunu göstermektedir.

UAOF, Uyarlanabilir Ağırlıklı Ortalama Filtre [40] filtresinde ise görüntü olarak yine Lena ile çalışıldığı için kıyaslama tablosuna yer verilmemiştir fakat çalışmada kullanılan 40 farklı test görüntüsünün ortalama sonuçlarına göre 0,2 yoğunlukta PSNR değeri FUMF [2] için 37,24, IAWMF [40] için 39,52 , 0,8 yoğunlukta PSNR değeri FUMF [2] için 26,41, IAWMF [40] için 28,27 olarak bulunmuştur.



Şekil 4.27 : (a) Orijinal görüntü, (b) 0,2 gürültülü görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.

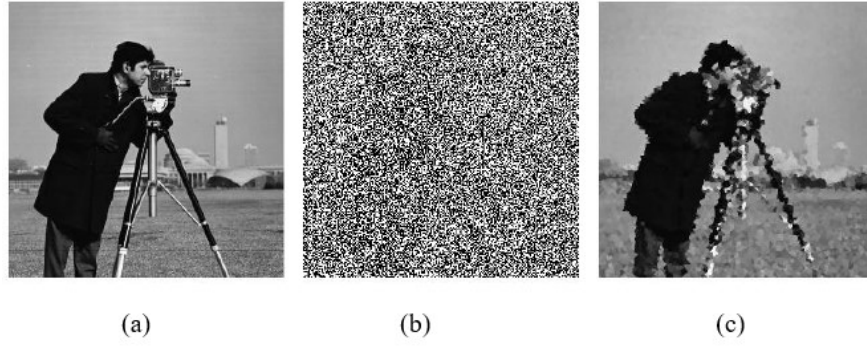


Şekil 4.28 : (a) Orijinal görüntü, (b) 0,5 gürültülü görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.

Bu sonuçlara dayanarak bizim elde ettiğimiz filtrelerden daha iyi sonuç verdiği çıkarımında bulunulabilir.

Ayrıntıya Duyarlı Filtre, ADF [41] filtresinde Kameraman görüntüsü kullanılmış ve Kameraman resminin PSNR değerleri 0,2 yoğunluğa sahip tuz ve biber gürültüsü için 32,62, 0,5 yoğunluğa sahip tuz ve biber gürültüsü için 26,93, 0,9 yoğunluğa sahip tuz ve biber gürültüsü için 21,78 olarak bulunmuştur. Aynı yoğunlukta Kameraman görüntüsünün YBMGMF9 algoritması ile elde edilen PSNR değerleri sırasıyla 32,17, 26,89 ve 20,68 'dir. Buradaki Kameraman görüntüsünün boyutu, önceki tablolarda kullanılan boyuttan farklı olduğu için PSNR değeri de farklıdır.

Karşılaştırılan filtreler arasında en iyi sonuç Farklı Uygulanabilir Medyan Filtre, FUMF [2] algoritmasına ait olduğundan, YBMGMF9 algoritmasıyla elde edilen filtre FUMF ile karşılaştırılmaktadır. Bu amaçla, Barbara, Bot, Ev, Köprü, Korsan ve Oturma Odası görüntüleri kullanılmış ve YBMG algoritması, 0,2, 0,4, 0,6 ve 0,8 yoğunlukları için tüm görüntülerde daha iyi sonuçlar vermektedir. Elde edilen PSNR değerleri Çizelge 4.18'te sunulmuş, SSIM değerleri ise Çizelge 4.19'te verilmiştir. Elaine görüntüsünün gürültülerden arındırılmış hali, FUMF ve YBMGMF kullanılarak



Şekil 4.29 : (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü.

Çizelge 4.18 : FUMF ve YBMGM9 filtrelerinin farklı yoğunluktaki PSNR değerleri.

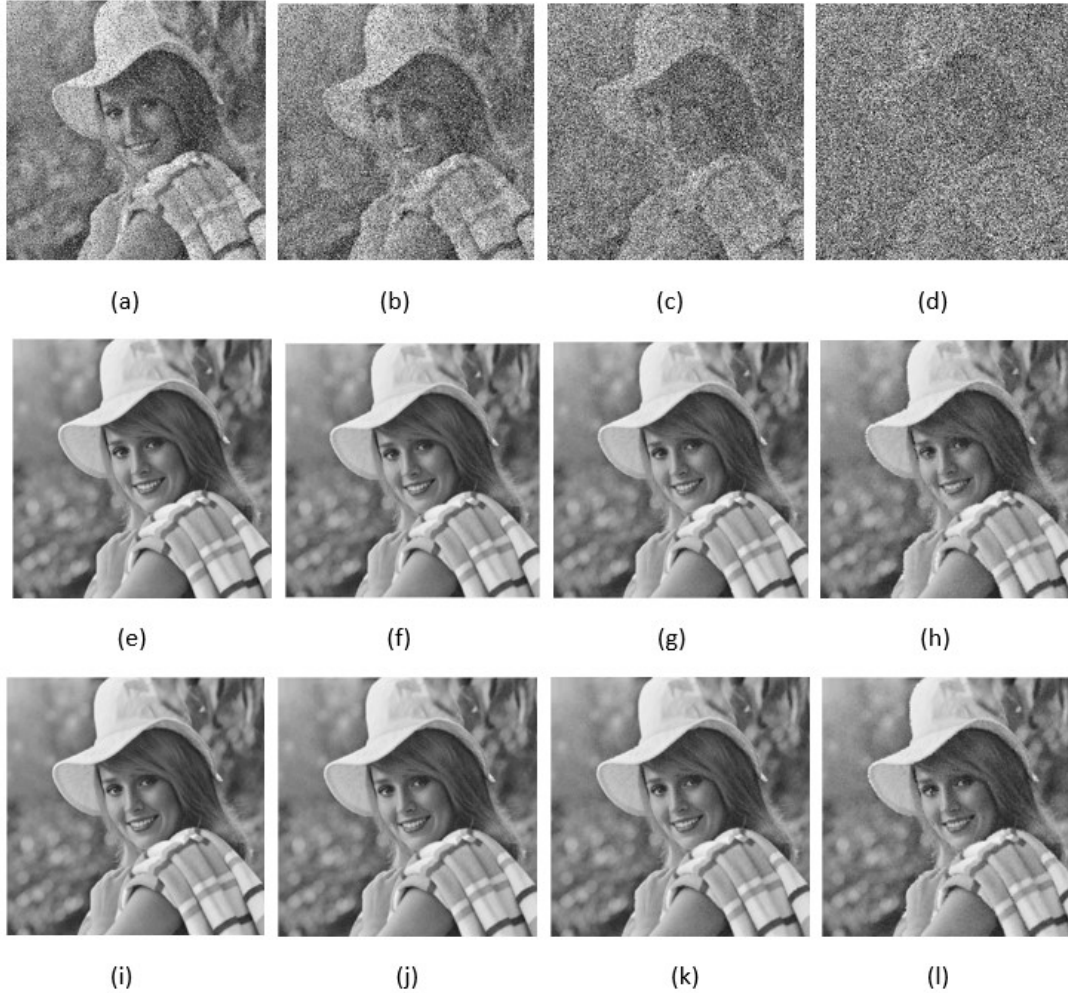
Görüntüler	Filtreler	0,2	0,4	0,6	0,8
Barbara	FUMF	30,50	26,91	24,54	22,37
	YBMGMF9	31,21	27,73	25,18	22,72
Bot	FUMF	35,21	31,17	28,29	25,42
	YBMGMF9	35,75	31,88	28,86	25,49
Ev	FUMF	36,51	32,58	29,95	27,15
	YBMGMF9	37,07	33,33	30,43	27,54
Köprü	FUMF	30,70	27,31	24,83	22,37
	YBMGMF9	31,32	28,13	25,42	22,65
Korsan	FUMF	33,78	30,03	27,28	24,37
	YBMGMF9	34,12	30,58	27,73	24,68
Oturma Odası	FUMF	35,04	30,97	28,05	25,16
	YBMGMF9	35,70	31,70	28,60	25,56

farklı yoğunluk seviyelerinde Şekil 4.30’da sunulmuştur [50]. Böylelikle YBMG ile tasarlanan YBMGMF9 filtresi, üretilen algoritmalar içinde en başarılı olan ve güncel çalışmaları geride bırakan çalışma olarak hem metrik değerleriyle hem de görüntülerle başarısını kanıtlamıştır.

2023 yılında yayımlanan Modifiye Edilmiş Farklı Uygulanabilir Medyan Filtre, MFUMF [35] (Modified Form of Different Applied Median Filter, MDAMF) makalesi, Table 4.20’te YBMGMF9 algoritması ile karşılaştırılmaktadır. Elde edilen sonuçlar, YBMG ile oluşturulan filtrenin, 0,2, 0,4, 0,6 ve 0,8 yoğunluk seviyelerinde Uçak, Köprü ve Bot görüntüleri için MFUMF’ye göre daha iyi PSNR değerleri sağladığını göstermektedir. Ayrıca, Lena görüntüsü, AGMF [38], KTA [36], NAFSM [39], YKTA, YKTA2 [37], MFUMF [35], UAOF [40] ve YBMGMF kullanılarak çeşitli yoğunluk seviyelerinde karşılaştırılmış ve YBMGMF, bu filtreler arasında tüm

Çizelge 4.19 : FUMF ve YBMGMF9 algoritmalarının farklı gürültü yoğunluklarındaki SSIM değerleri.

Görüntüler	Filtreler	0,2	0,4	0,6	0,8
Barbara	FUMF	0,9586	0,9025	0,8263	0,7132
	YBMGMF9	0,9633	0,9148	0,8389	0,7240
Bot	FUMF	0,9627	0,9135	0,8457	0,7381
	YBMGMF9	0,9661	0,9217	0,8581	0,7473
Ev	FUMF	0,9650	0,9174	0,8513	0,7472
	YBMGMF9	0,9683	0,9259	0,8627	0,7546
Köprü	FUMF	0,9474	0,8816	0,7899	0,6451
	YBMGMF9	0,9514	0,8916	0,8027	0,6516
Korsan	FUMF	0,9536	0,9044	0,8355	0,7196
	YBMGMF9	0,9563	0,9117	0,8430	0,7285
Oturma Odası	FUMF	0,9663	0,9177	0,8466	0,7333
	YBMGMF9	0,9692	0,9265	0,8597	0,7414



Şekil 4.30 : Elaine görüntüsünde, sırasıyla (a),(b),(c) ve (d) de 0,2, 0,4, 0,6 ve 0,8 yoğunluklarda tuz ve biber gürültüsü uygulanmış, (e), (f), (g) ve (h) de FUMF kullanılarak, (i), (j), (k) ve (l) de YBMGMF9 filtresi kullanılarak temizlenmiş görüntüler

Çizelge 4.20 : MFUMF ve YBMGMF9 algoritmalarının farklı gürültü yoğunluklarındaki PSNR değerleri.

Görüntüler	Filtreler	0,2	0,4	0,6	0,8
Uçak	KTA	27,53	25,08	22,02	18,62
	YKTA	29,66	27,95	24,87	20,84
	YKTA2	30,53	29,03	26,44	21,97
	FUMF	36,97	32,48	29,34	26,27
	MFUMF	37,43	32,39	29,19	26,18
	YBMGMF9	38,34	33,85	30,30	26,62
Köprü	KTA	24,38	22,27	20,13	17,50
	YKTA	27,26	25,05	22,41	20,13
	YKTA2	30,00	26,70	23,78	20,17
	FUMF	30,35	26,88	24,51	22,12
	MFUMF	30,71	27,18	24,67	22,25
	YBMGMF9	31,12	28,04	25,46	22,71
Bot	KTA	27,60	25,13	22,36	19,00
	YKTA	31,48	28,37	25,00	21,16
	YKTA2	35,30	30,90	27,01	22,53
	FUMF	35,29	31,23	28,34	25,36
	MFUMF	35,34	31,19	28,32	25,51
	YBMGMF9	35,83	31,81	28,73	25,65

test edilen yoğunluk seviyelerinde en iyi sonuçları sağlamıştır. Sonuçlar Çizelge 4.21’de sunulmuştur.

YBMGMF9 algoritmasının PSNR ve SSIM metriklerinden farklı olarak MS-SSIM metriğiyle de başarısı hesaplanmış ve elde edilen sonuçlar 4.22’de verilmiştir.

YBMGMF9 ve ÇYÇGMF5 filtrelerinin renkli görüntülerden elde edilen PSNR ve SSIM değerleri Çizelge 4.23, Çizelge 4.24 ve Çizelge 4.25 ’de verilmiştir. Gürültüleri giderilen görüntüler ise Şekil 4.31, Şekil 4.32 , Şekil 4.33, Şekil 4.34 ve Şekil 4.35’de

Çizelge 4.21 : Farklı gürültü seviyeleri altında Lena görüntüsü için PSNR değerlerinin karşılaştırılması.

Filtreler	0,2	0,4	0,6	0,8
AGMF	32,98	24,92	12,30	8,16
KTA	33,39	27,05	21,86	16,45
NAFSM	35,65	32,39	29,91	27,14
YKTA	35,41	31,64	27,67	23,12
YKTA2	39,37	34,46	29,98	24,61
MFUMF	39,31	34,87	31,84	28,61
UAF	39,80	35,84	32,62	29,32
YBMGMF9	40,73	36,67	33,40	29,90

Çizelge 4.22 : YBMGMF9 filtresinin farklı yoğunluklardaki MS-SSIM değerleri.

Görüntüler	0,2	0,4	0,6	0,8
Barbara	0,9893	0,9736	0,9495	0,9068
Bot	0,9947	0,9864	0,9710	0,9330
Ev	0,9952	0,9873	0,9735	0,9377
Köprü	0,9897	0,9774	0,9540	0,8969
Korsan	0,9937	0,9852	0,9703	0,9326
Oturma Odası	0,9951	0,9871	0,9712	0,9308

verilmiştir. Renkli görüntülerde literatürde çok geniş çalışma bulunmaması nedeniyle sonuçlar herhangi başka yöntemlerle kıyaslanmamıştır fakat görseller incelendiğinde hem düşük hem de yüksek yoğunluk seviyelerinde üstün performans gözlemlenmiştir.

Çizelge 4.23 : 0,2 yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntülerin PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	YBMGMF9	ÇYÇGMF5
Ev	PSNR	36,9705	36,5544
	SSIM	0,9832	0,9818
Barbara	PSNR	34,2440	34,0229
	SSIM	0,9778	0,9770
Baboon	PSNR	28,8885	28,6267
	SSIM	0,9642	0,9623
Biber	PSNR	36,1621	35,5541
	SSIM	0,9953	0,9948
Göl	PSNR	31,3299	31,0256
	SSIM	0,9816	0,9805

Çizelge 4.24 : 0,5 yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntülerin PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	YBMGMF9	ÇYÇGMF5
Ev	PSNR	31,7794	31,6759
	SSIM	0,9442	0,9429
Barbara	PSNR	29,3439	29,3535
	SSIM	0,9289	0,9284
Baboon	PSNR	24,3474	24,2721
	SSIM	0,8890	0,8874
Biber	PSNR	32,1631	31,8844
	SSIM	0,9885	0,9878
Göl	PSNR	27,8151	27,7329
	SSIM	0,9501	0,9492

YBMG ve ÇYÇG ile daha önce elde edilen algoritmalar hem düşük yoğunlukta hem yüksek yoğunlukta çok iyi sonuçlar elde edilmiştir. Yöntemler siyah beyaz görüntüler

Çizelge 4.25 : 0,9 yoğunluklu tuz ve biber gürültüsüne sahip renkli görüntülerin PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	YBMGMF9	ÇYÇGMF5
Ev	PSNR	25,3538	25,2789
	SSIM	0,7869	0,7852
Barbara	PSNR	23,5022	23,5136
	SSIM	0,7794	0,7801
Baboon	PSNR	19,1351	19,1329
	SSIM	0,6520	0,6530
Biber	PSNR	25,5404	25,5010
	SSIM	0,9502	0,9495
Göl	PSNR	21,8919	21,8837
	SSIM	0,8290	0,8298



(a)

(b)

(c)

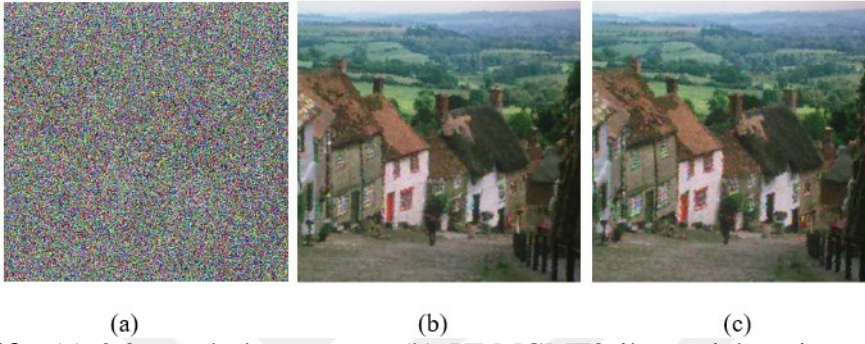
Şekil 4.31 : (a) 0,2 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü

üzerinde farklı yoğunluk değerlerinde literatürün en iyi filtrelerinden olan FUMF ile kıyaslandığında daha iyi sonuçlar elde edilmiştir. Hem PSNR hem de SSIM değerleri daha yüksektir. YBMG ve ÇYÇG kendi arasında kıyaslandığında ise YBMG düşük yoğunluk değerlerinde çok az farklarla ÇYÇG'den daha iyi sonuç vermekle beraber yoğunluk yükseldiğinde ÇYÇG'nin daha iyi sonuç verdiği görüntüler de olmuştur. Farklı yoğunluk değerlerinde FUMF ile kıyaslanarak elde edilen sonuçlar Çizelge 4.26'de verilmiştir. YBMG, ÇYÇG ve FUMF ile temizlenen görüntüler ise Şekil 4.36 ve Şekil 4.37'de verilmiştir. Deneysel sonuçlar göstermiştir ki YBMG ve ÇYÇG ile elde edilen tuz ve biber gürültüsünü giderme algoritmaları istatistiksel yöntemlerin birçoğundan daha iyi sonuç vermiştir.

Filtreler renkli görüntüler için de uyarlanmış burada da görsel olarak gürültülerden çok iyi temizlendiği görülmüştür fakat literatür araştırıldığında yapılan çalışmaların daha çok siyah beyaz olması dolayısıyla rekli görüntüler için farklı resimler üzerinde,



Şekil 4.32 : (a) 0,5 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü



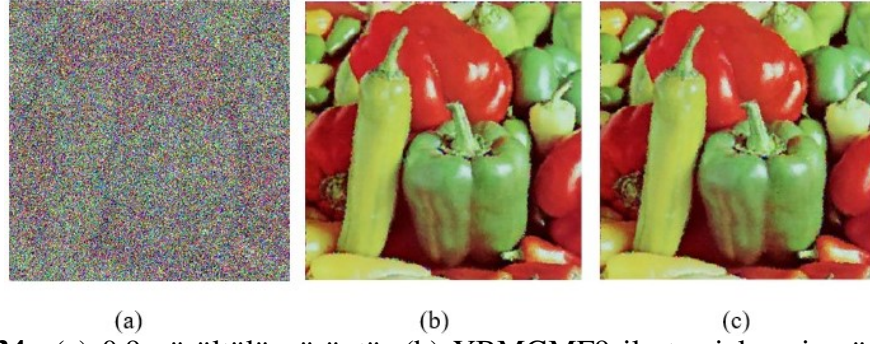
Şekil 4.33 : (a) 0,9 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü

farklı yoğunluk değerlerinde kıyaslama yapılabilecek geniş bir arşiv bulunamadığı için sonuçlar direkt verilmiştir.

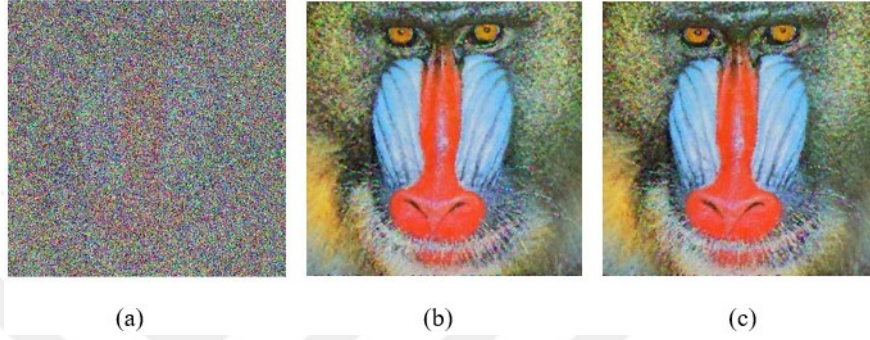
Çizelge 4.26 : 0,9 yoğunluklu tuz ve biber gürültüsüne sahip görüntülerin FUMF, YBMGMF9 VE CYCG5 ile elde edilen PSNR ve SSIM değerleri.

Görüntüler	Ölçüler	FUMF	YBMGMF9	ÇYÇGMF5
Ev	PSNR	25,0893	25,4904	25,3384
	SSIM	0,6471	0,6576	0,6548
Kameraman	PSNR	24,8601	24,8883	25,0047
	SSIM	0,8303	0,8318	0,8343
Oturma Odası	PSNR	23,0367	23,2899	23,2947
	SSIM	0,6210	0,6285	0,6277
Barbara	PSNR	21,0069	21,2752	21,2329
	SSIM	0,6162	0,6254	0,6261
Monarch	PSNR	19,9543	20,0721	20,0273
	SSIM	0,7156	0,7139	0,7150

Tablo 4.27’de verilen PSNR değerlerine bakıldığında YBMG ile elde edilen filtrelerin tamamı içinde 0,1 yoğunlukta en iyi sonuç YBMGMF9 algoritmasına aittir. Ayrıca YBMGMF9 ve literatürün en iyi çalışmalarından FUMF algoritmalarının düşük yoğunluktan yüksek yoğunluğa kadar PSNR değerlerinin nasıl değiştiği Şekil



Şekil 4.34 : (a) 0,9 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü

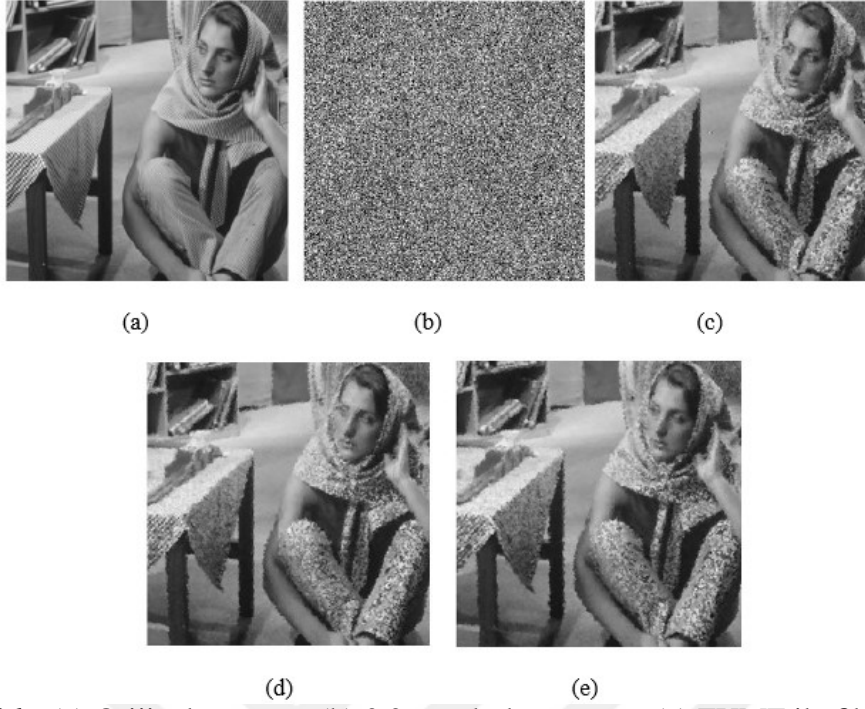


Şekil 4.35 : (a) 0,9 gürültülü görüntü, (b) YBMGMF9 ile temizlenmiş görüntü, (c) ÇYÇGMF5 ile temizlenmiş görüntü

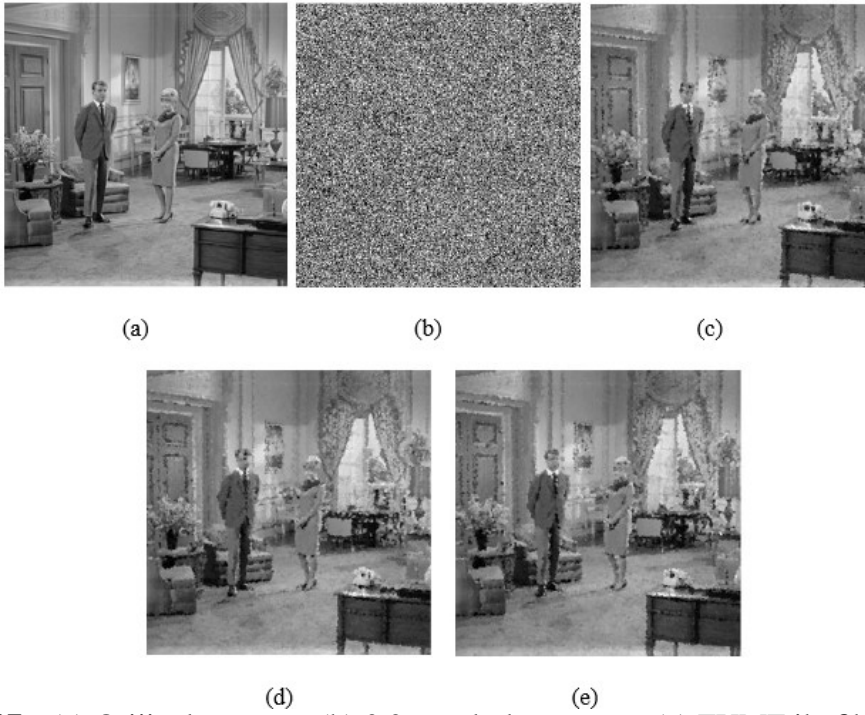
Çizelge 4.27 : YBMG ile elde edilen algoritmaların 0,1 yoğunluktaki tuz ve biber gürültüsüne sahip görüntüler için PSNR değerleri.

Filtreler	Kameraman	Barbara	Oturma Odası	Monarch
YBMGMF1	20,0001	19,2674	20,6823	24,0480
YBMGMF2	31,7743	31,0528	32,2904	33,0324
YBMGMF3	19,3945	19,6459	20,0935	19,8948
YBMGMF4	32,3944	31,7489	33,4223	33,7985
YBMGMF5	30,0883	30,2584	31,8046	30,7324
YBMGMF6	34,5049	31,9951	33,5684	36,0985
YBMGMF7	44,3350	34,3371	38,8850	37,6011
YBMGMF8	44,1538	34,4585	38,5903	37,2372
YBMGMF9	44,2138	34,5754	38,9042	37,3055

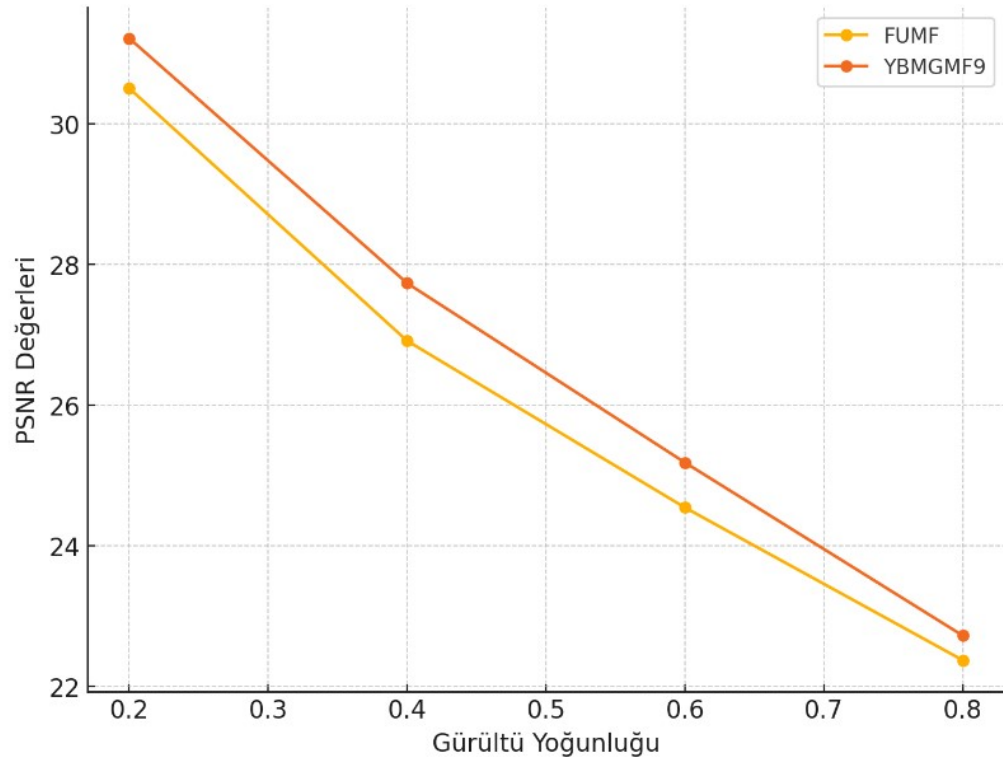
4.38 'deki garfikte verilmiştir. Buna göre her yoğunluk seviyesinde YBMGMF9 algoritmasının FUMF'den daha iyi sonuç verdiği bir kez daha görülmüştür.



Şekil 4.36 : (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) FUMF ile filtrelenmiş, (d) YBMGMF9 ile filtrelenmiş, (e) ÇYÇGMF5 ile filtrelenmiş



Şekil 4.37 : (a) Orijinal görüntü, (b) 0,9 gürültülü görüntü, (c) FUMF ile filtrelenmiş, (d) YBMGMF9 ile filtrelenmiş, (e) ÇYÇGMF5 ile filtrelenmiş



Şekil 4.38 : FUMF ve YBMGMF9 filtrelerinin farklı yoğunluk değerlerindeki PSNR değerleri.



5. SONUÇLAR

Bu çalışmada, gri tonlu ve renkli görüntülerden tuz ve biber gürültüsünü temizlemek için yeni filtreler tasarlanmıştır. Bu filtrelerin tasarımında YBMG ve ÇYÇG algoritmaları kullanılmıştır ve bu, YBMG'nin ve ÇYÇG'nin görüntü filtreleme alanında kullanıldığı ilk örnektir. Öncelikle Gauss ve alacalı gürültü eklenmiş görüntülere bakıldığında ÇYÇGMF2 algoritmasının en temel filtreler olan ortalama ve medyan filtreden daha iyi sonuç verdiği Çizelge 4.1 ve 4.2'de görülmüştür. Tuz ve biber gürültüsünü gidermek için en temel filtre olan medyan filtre düşük yoğunluklarda ilk kez YBMGMF2 algoritması ile geçilmiş ve çalışmanın devamında bu durum iyileştirilmeye çalışılmıştır. YBMGMF5 algoritması iteratif olmayan bir yöntem sunarken yine tuz ve biber gürültüsünü düşük yoğunluklarda medyan filtreden daha iyi gidermiştir. Buraya kadar elde edilen sonuçlar düşük yoğunluklarda medyan filtreden iyi olmakla beraber literatürün güncel çalışmalarıyla kıyaslandığında özellikle yüksek yoğunluklarda geride kalmıştır. Sonuçları iyileştirmek için tekrarsız medyan filtreden ve kullanılan pencere içindeki standart sapma değerinden faydalanan hibrit bir model olan YBMGMF6 filtresi üretilmiştir ancak bu yöntemde de çok büyük bir iyileşme ortaya çıkmamıştır. Pencere boyutunu büyütmenin yüksek yoğunluk derecelerinde işe yarayacağı varsayımıyla 3x3 lük pencere boyutları simetrileri yardımıyla 9x9 boyutuna çıkarılmış ve böylelikle YBMGMF7 algoritması oluşturulmuştur. Çizelge 4.9' de gözlemlendiği gibi 0,2 yoğunlukta kıyaslanan filtrelerden daha iyi sonuçlar elde edilmiştir ancak yine de yüksek yoğunluklarda istenen değerlere ulaşılamamıştır. Daha iyi sonuçlara ancak dinamik pencere boyutu kullanıldığında ulaşılacağı tahmininden yola çıkarak YBMGMF9 algoritması oluşturulmuş ve Çizelge 4.15'de görüldüğü gibi en iyi sonuçlar elde edilmiştir ve YBMG medyan filtresi olarak [50]'da verilen makale olarak yayınlanmıştır.

Gürültü temizleme potansiyelinin yüksek olduğu düşünülerek ÇYÇG ile de benzer filtreler elde edilmiş ve Çizelge 4.17'de görüldüğü gibi ÇYÇG ile de özellikle yüksek yoğunluklarda başarılı sonuçlar elde edilmiştir.

YBMG ve ÇYÇG ile üretilen algoritmalar kendi arasında kıyaslandığında gri ton görüntülerde YBMG algoritmasının daha başarılı olduğu gözlemlenirken, renkli görüntülerde ve özellikle yüksek yoğunluklarda ÇYÇG algoritmasının daha iyi sonuçlar verdiği görülmüştür.

Öğrenme tabanlı yöntemler, görüntü üzerindeki gürültüleri giderme konusunda genellikle istatistiksel yöntemlere göre daha başarılıdır. Bu durum, öğrenme tabanlı yöntemlerin görüntülerdeki karmaşık desenleri ve yapı taşlarını daha iyi öğrenebilme yeteneklerinden kaynaklanmaktadır. Özellikle derin öğrenme teknikleri, çok katmanlı sinir ağları sayesinde görüntülerdeki ince detayları ve gürültüleri etkili bir şekilde ayırt edebilir. Bu bağlamda, derin öğrenme temelli yöntemler, büyük veri setleri üzerinde eğitilerek yüksek doğruluk oranlarına ulaşabilir ve daha az hata payı ile çalışabilirler.

Ancak, öğrenme tabanlı yöntemlerin istatistiksel yöntemlere kıyasla bazı belirgin avantajları ve dezavantajları bulunmaktadır. Öğrenme tabanlı yöntemlerin en büyük avantajlarından biri, çok büyük veri setleri üzerinde eğitildiğinde çok yüksek performans göstermeleridir. Bu yöntemler, veri miktarı arttıkça daha da iyi sonuçlar verebilir ve çeşitli görüntüleme koşullarına uyum sağlayabilirler. Bununla birlikte, bu yöntemlerin etkin bir şekilde çalışabilmesi için büyük miktarda etiketli veri gerekmektedir. Büyük veri setlerinin toplanması, işlenmesi ve saklanması ciddi bir lojistik ve maliyet gerektirir.

Öğrenme tabanlı yöntemler ayrıca yüksek hesaplama gücü ve performanslı donanımlara ihtiyaç duyarlar. Bu durum, özellikle sınırlı kaynaklara sahip kuruluşlar için bir dezavantaj oluşturabilir. Ayrıca, bu yöntemlerin eğitim süreci uzun zaman alabilir ve yüksek enerji tüketimine neden olabilir.

Bizim yöntemimiz ise istatistiksel bir yöntem olarak öğrenme tabanlı yöntemlerin gerisinde kalabilir; ancak bu yöntemlerin bazı kısıtlamalarına sahip değildir. İstatistiksel yöntemler, genellikle daha az veri ile çalışabilir ve büyük veri setlerine ihtiyaç duymazlar. Bu da veri toplama ve işleme maliyetlerini düşürür. Ayrıca, istatistiksel yöntemler genellikle daha düşük hesaplama gücü gerektirir ve bu nedenle ortalama performansa sahip makinelerde bile uygulanabilir. Bu durum, maliyet açısından büyük bir avantaj sağlar ve geniş bir kullanıcı kitlesi tarafından erişilebilir hale gelir.

Oluşturduğumuz algoritma, çevresel etkilere ihtiyaç duymadan hızlı bir şekilde uygulanabilir. Matematiksel temellere dayandığı için sonuçları kolayca yorumlanabilir ve analiz edilebilir. İstatistiksel yöntemlerin sağlam temelleri, algoritmanın tutarlı ve öngörülebilir sonuçlar vermesini sağlar. Bu yöntem, belirli durumlarda öğrenme tabanlı yöntemlere göre daha pratik ve ekonomik bir çözüm sunabilir.

Özetle, öğrenme tabanlı yöntemler genel olarak daha yüksek performans gösterebilir; ancak istatistiksel yöntemler, yeterli iyileştirmeyi sağlayabildikleri örneklemelerde avantajlı olabilir. İstatistiksel yöntemlerin düşük maliyetli, kolay uygulanabilir ve geniş erişilebilirlik özellikleri, bu yöntemleri belirli uygulamalar için cazip hale getirir. Bu nedenle, her iki yaklaşımın da kendine özgü güçlü ve zayıf yönleri bulunmaktadır ve bu yöntemlerin seçimi, uygulamanın gereksinimlerine ve kaynaklarına bağlı olarak değişebilir.



KAYNAKLAR

- [1] **Çelik, K., Sayan, H.H. ve Demirci, R.** (2018). Gradyan Uyarlamalı Gauss Görüntü Filtresi, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 6(1), 196–215.
- [2] **Erkan, U., Gökrem, L. ve Enginoğlu, S.** (2018). Different applied median filter in salt and pepper noise, *Computers and Electrical Engineering*, 70, 789–798.
- [3] **Singh, P. ve Shree, R.** (2016). Speckle Noise: Modelling and Implementation, *International Science Press*, 9(17), 8717–8727.
- [4] **Lee, D., Luk, W., Villasenor, J. ve Cheung, P.** (2004). A Gaussian noise generator for hardware-based simulations, *IEEE Transactions on Computers*, 53(12), 1523–1534.
- [5] **Patidar, P. ve Dadheech, P.** (2019). Performance of Fuzzy Filter and Mean Filter for Removing Gaussian Noise, *International Journal of Computer Applications*, 182(38), 29–35.
- [6] **Mukesh, C., Mukesh, C. ve Rakhi, C.** (2004). Survey of Image Denoising Techniques, Santa Clara, s.27–30.
- [7] **Thivakaran, T. ve Chandrasekaran, M.** (2010). Nonlinear Filter Based Image Denoising Using AMF Approach, *International Journal of Computer Science and Information Security*, 7(2).
- [8] **Ville, D., Nachtegaal, M. ve Weken, D.** (2000). New fuzzy filter for Gaussian noise reduction, *Visual Communications and Image Processing*.
- [9] **Küpeli, C. ve Bulut, F.** (2020). Görüntüdeki Tuz Biber ve Gauss Gürültülerine Karşı Filtrelerin Performans Analizleri, *Haliç Üniversitesi Fen Bilimleri Dergisi*, 3(2), 211–239.
- [10] **Lee, J.** (1981). Refined filtering of image noise using local statistics, *Computer Graphics and Image Processing*, 15(4), 380–389, <https://www.sciencedirect.com/science/article/pii/S0146664X81800184>.
- [11] **Kuan, D., Sawchuk, A., Strand, T. ve Chave, P.** (1985). Adaptive Noise Smoothing Filter for Images with Signal-Dependent Noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7, 165–177.
- [12] **Frost, V., Stiles, J., Shanmugan, K. ve Holtzman, J.** (1982). A model for radar images and its application to adaptive digital filtering of multiplicative

noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4, 157–166.

- [13] **Lopes, A., Touzi, R. ve Nezry, E.** (1990). Adaptive speckle filters and scene heterogeneity, *IEEE Transactions on Geoscience and Remote Sensing*, 28(6), 992–1000.
- [14] **Abd, E., Youssef, A. ve Kadah, Y.** (2002). Real-time speckle reduction and coherence enhancement in ultrasound imaging via nonlinear anisotropic diffusion, *IEEE Transactions on Biomedical Engineering*, 49(9), 997–1014.
- [15] **Justusson, B.** (1981). Median filtering: statistical properties. In: *Two-dimensional digital signal processing, II. Springeri*, 161–196.
- [16] **Ibrahim, H., Kong, N. ve Ng, T.** (2008). Simple adaptive median filter for the removal of impulse noise from highly corrupted images., *IEEE Trans Consum Electron*, 54(4), 1920–1927.
- [17] **Erkan, U., Thanh, D., Enginoğlu, S. ve Memiş, S.** (2020). Improved Adaptive Weighted Mean Filter for Salt and Pepper Noise Removal, *ICECCE*, İstanbul, Turkey.
- [18] **Bai, T., Tan, J., Hua, M. ve Wang, Y.** (2014). A novel algorithm for removal of salt and pepper noise using continued fractions interpolation, *Signal Processing*, 102, 247–255.
- [19] **Li, B. ve Xie, W.** (2016). Image denoising and enhancement based on adaptive fractional calculus of small probability strategy, *Neurocomputing*, 175, 704–714.
- [20] **Erkan, U. ve Gökrem, L.** (2018). A new method based on pixel density in salt and pepper noise removal, *Turkish Journal of Electrical Engineering and Computer Sciences*, 26, 162–171.
- [21] **Anil, K., Rajesh, K., Pandey, P. ve Reddy, K.** (2020). Generalized fractional derivative based adaptive algorithm for image denoising, *Multimedia Tools and Applications*, 79(1), 14201–14224.
- [22] **Garg, B. ve Arya, K.** (2020). Four stage median-average filter for healing high density salt and pepper noise corrupted images., *Multimedia Tools Applications*, 79, 32305–32329.
- [23] **Karthik, B. ve Krishna, K.** (2021). Removal of high density salt and pepper noise in color image through modified cascaded filter, *Ambient Intell Human Comput*, 12, 3901–3908.
- [24] **Thanh, D., Hai, N. ve Prasath, V.** (2020). A two-stage filter for high density salt and pepper denoising, *Multimedia Tools and Applications*, 79, 21013–21035.
- [25] **Chan, R., Ho, C. ve Nikolova, M.** (2005). Salt and Pepper Noise Removal by Median Type Noise Detectors and Detail Preserving Regularization., *IEEE Transactions on Image Processing*, 14(10), 1479 – 1485.

- [26] **Esakkirajan, S., Veerakumar, T., Subramanyam, A.N. ve PremChand, C.H.** (2011). Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter., *IEEE Signal Processing Letters*, 18(5), 287–290.
- [27] **Banerjee, S., Debarpito, S. ve Chatterjee, D.** (2021). High-Density Salt and Pepper Noise Removal from Colour Images by Introducing New Enhanced Filter, *2021 International Conference on Intelligent Technologies (CONIT)*, s.1–6.
- [28] **Zaini, H.** (2021). Window Median Filter (WMF) to Eliminate Salt and Pepper Noise from Digital Color Image, *International Journal of Computer Science and Mobile Computing*, 10(9), 23–37.
- [29] **Zhang, H., Zhu, Y. ve Zheng, H.** (2021). NAMF: A Nonlocal Adaptive Mean Filter for Removal of Salt-and-Pepper Noise, *Mathematical Problems in Engineering*.
- [30] **Enginoğlu, S., Erkan, U. ve Memiş, S.** (2022). Exponentially Weighted Mean Filter for Salt-and-Pepper Noise Removal, *Artificial Intelligence in Data and Big Data Processing*, cilt124.
- [31] **Vasanth, K. ve Varatharajan, R.** (2020). An adaptive content based closer proximity pixel replacement algorithm for high density salt and pepper noise removal in images., *Ambient Intell Human Comput.*
- [32] **Chen, F., Huang, M., Ma, Z., Li, Y. ve Huang, Q.** (2020). An Iterative Weighted-Mean Filter for Removal of High-Density Salt-and-Pepper Noise, *Symmetry*, 12(12).
- [33] **Amiri, S.** (2020). Salt and Pepper Noise Removal using Pixon-based Segmentation and Adaptive Median Filter, *Journal of AI and Data Mining*, 8(1), 119–126.
- [34] **Hien, N., Thanh, D., Erkan, U., Tavares, J. ve Manuel, R.** (2022). Image Noise Removal Method Based on Thresholding and Regularization Techniques, *IEEE Access*, 10, 71584–71597.
- [35] **Aslam, N., Ehsan, M., Rehman, Z., Hanif, M. ve Mustafa, G.** (2023). A modified form of different applied median filter for removal of salt and pepper noise, *Multimedia Tools and Applications*, 82(5), 7479–7490.
- [36] **Pattnaik, A. ve Agarwal, S.** (2008). A new and efficient method for removal of high density salt and pepper noise through cascade decision based filtering algorithm, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 19(21).
- [37] **Revathy, K., Madhu, R. ve Nair, S.** (2008). Removal of salt-and-pepper noise in images: A new decision based algorithm, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 19(21).

- [38] **Whang, Z. ve Zhang, D.** (1999). Progressive switching median filter for the removal of impulse noise from highly corrupted images, *IEEE Transactions on Circuits and Systems*, 46(1), 78–80.
- [39] **Toh, K. ve Isa, N.** (2010). Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction, *IEEE Signal Process Letters*, 17(3), 281–4.
- [40] **Erkan, U., Thanh, D., Enginoğlu, S. ve Memiş, S.** (2020). Improved Adaptive Weighted Mean Filter for Salt and-Pepper Noise Removal, Istanbul.
- [41] **Liang, H., Li, N. ve Zhao, S.** (2021). Salt and Pepper Noise Removal Method Based on a Detail-Aware Filter, *Symmetry*, 13(3), <https://www.mdpi.com/2073-8994/13/3/515>.
- [42] **Xing, Y., Xu, J. ve Tan, J.** (2019). Deep CNN for removal of salt and pepper noise, *IET Image Processing*, 13(9), 1550–1560.
- [43] **Sobol, I.** (1993). Sensitivity estimates for nonlinear mathematical models, *Mathematical modelling and computational experiments*, 1(4), 407–414.
- [44] **Gao, S. ve Gruev, V.** (2011). A new approach for data partitioning through high dimensional model representation, *Optics Express*, 19(27), 26161–26173.
- [45] **Demiralp, M.** (2003). High Dimensional Model Representation and Its Application Varieties, *Mathematical Research*, 9(1), 146–159.
- [46] **Tunga, B. ve Koçanaoğulları, A.** (2017). Digital image decomposition and contrast enhancement using high-dimensional model representation, *Signal, Image and Video Processing*, 12(1), 299–306.
- [47] **Tunga, M.** (2014). A new approach for multivariate data modelling in orthogonal geometry, *International Journal of Computer Mathematics*, 92(9), 2011–2021.
- [48] **Okan, A., Baykara, N. ve Demiralp, A.** (2010). Weight optimization in Enhanced Multivariate Product Representation (EMPR) method, *International conference of numerical analysis and applied mathematics*, AIP Conf. Proc., s.1935–1938.
- [49] **Tuna, S. ve Tunga, B.** (2013). A novel piecewise multivariate function approximation method via universal matrix representation, *Journal of Mathematical Chemistry*, 51, 1784–1801.
- [50] **Kacar, S. ve Tunga, B.** (2024). High Dimensional Model Representation Median Filter, *Signal Image and Video Processing for Removing Salt and Pepper Noise*.
- [51] **Nasr, M. ve Alrahmawy, M.F.** (2017). Multi-scale structural similarity index for motion detection, *Journal of King Saud University - Computer and Information Sciences*, 29(3), 399–409.

ÖZGEÇMİŞ

Ad Soyad : Sena KAÇAR

EĞİTİM :

- **Lisans** : 2012, İstanbul Teknik Üniversitesi, Fen Edebiyat Fakültesi, Matematik Mühendisliği
- **Yüksek Lisans** : 2018, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Mühendisliği

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Kacar S, Tunga B** 2024. High Dimensional Model Representation Median Filter for Removing Salt and Pepper Noise , *Signal Image and Video Processing*

F. M. SURNAME

YÜKSEK BOYUTLU MODEL GÖSTERİLİMİ ve ÇOK DEĞİŞKENLİLİĞİ YÜKSELTİLMİŞ
ÇARPIMLAR GÖSTERİLİMİ ile GÖRÜNTÜ ÜZERİNDEKİ GÜRÜLTÜLERİ GİDERME

2020