

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UZAK MESAFELİ GÖRÜNTÜLEMEDE BOZULMALARIN
İNCELENMESİ VE İYİLEŞTİRİLMESİ

Gülsüm GEZER

DOKTORA TEZİ

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı
Kontrol ve Otomasyon Mühendisliği Programı

Danışman

Dr. Öğr. Üyesi Muharrem MERCİMEK

Şubat, 2020

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UZAK MESAFELİ GÖRÜNTÜLEMEDE BOZULMALARIN
İNCELENMESİ VE İYİLEŞTİRİLMESİ

Gülsüm GEZER tarafından hazırlanan tez çalışması 25.02.2020 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Kontrol ve Otomasyon Mühendisliği Anabilim Dalı Kontrol ve Otomasyon Mühendisliği Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Muharrem MERCİMEK
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Muharrem MERCİMEK, Danışman
Yıldız Teknik Üniversitesi

Prof. Dr. Şeref Naci ENGİN, Üye
Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Janset DAŞDEMİR, Üye
Yıldız Teknik Üniversitesi

Prof. Dr. Bahadır Kürşat GÜNTÜRK, Üye
Medipol Üniversitesi

Prof. Dr. Galip CANSEVER, Üye
Altınbaş Üniversitesi

Danışmanım Dr. Öğr. Üyesi Muharrem MERCİMEK sorumluluğunda tarafımda hazırlanan Uzak Mesafeli Görüntüleme Bozulmaların İncelenmesi ve İyileştirilmesi başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Gülsüm GEZER

İmza



Bu çalışma, TÜBİTAK tarafından 2214-A programı kapsamında desteklenmiştir.

Aileme ve beni yetiřtiren saygıdeęer retmenlerime...



TEŞEKKÜR

Doktora öğrenimim süresince hoşgörüsü ve desteği ile yanımda olan danışmanım Sayın Dr. Öğr. Üyesi Muharrem MERCİMEK'e teşekkür ederim. Ayrıca değerli görüşleriyle her zaman bana yol gösteren hocalarım Sayın Prof. Dr. Bahadır Kürşat GÜNTÜRK, Doç. Dr. Akın DELİBAŞI ve Dr. Öğr. Üyesi Janset DAŞDEMİR'e de teşekkürü bir borç bilirim.

Doktora tez çalışmam süresince verdiği donanımsal destekten dolayı Sayın Doç. Dr. Türker TÜRKER'e minnettarım. Manevi desteklerinden dolayı dostlarım Doç. Dr. Yasemin ÖNER, Dr. Öğr. Üyesi Fatma YILDIZ TAŞCIKARAOĞLU, Ayşenur YÜKSEL, Ayşe ÖZÇELİK, Halit ÖRENBAŞ ve adını sayamadığım diğer dostlarıma sonsuz teşekkürlerimi sunarım.

Tokyo Teknoloji Enstitüsü, Sistem ve Kontrol Mühendisliği departmanında Fujita-Hatanaka Laboratuvarında 11 ay boyunca araştırma yapmama olanak sağlayan ve tezime büyük katkısı olan TÜBİTAK'a teşekkür ederim.

Doktora tezimin ve hayatımın her aşamasında yanımda olup, beni koruyup kolladığını bildiğim, varlığını her zaman hissettiğim canım abim Ferit GEZER'e ve aileme minnettarım.

Son teşekkürümü de öğretim hayatım boyunca karşıma çıkıp bana değer katan bütün öğretmenlerime yapmak isterim. İyi ki vardınız, iyi ki varsınız.

Gülsüm GEZER

İÇİNDEKİLER

| | |
|---|-------------|
| SİMGE LİSTESİ | viii |
| KISALTMA LİSTESİ | xi |
| ŞEKİL LİSTESİ | xii |
| TABLO LİSTESİ | xvi |
| ÖZET | xvii |
| ABSTRACT | xix |
| 1 GİRİŞ | 1 |
| 1.1 Literatür Özeti | 1 |
| 1.2 Tezin Amacı | 7 |
| 1.3 Orjinal Katkı | 8 |
| 1.4 Motivasyon | 8 |
| 1.5 Tezin İçeriği | 9 |
| 2 UZAK MESAFELİ GÖRÜNTÜLEMEDE İMGELERİN ELDE EDİLMESİ VE OLUŞAN BOZULMALAR | 10 |
| 2.1 Giriş | 10 |
| 2.2 Kolmogorov Türbülans Teorisi | 12 |
| 2.3 İmgelerin Elde Edilmesi | 14 |
| 2.4 İmgelerdeki Bozulmaların Piksel Doluluğu Modelleme Kullanılarak Analizi | 21 |
| 2.4.1 Deneysel Sonuçlar | 23 |
| 2.5 İmgelerdeki Bozulmaların Gauss Karışım Modeli Kullanılarak Analizi . | 32 |
| 2.5.1 Deneysel Sonuçlar | 36 |
| 3 UZAK MESAFELİ GÖRÜNTÜLEMEDE İMGE ÇAKIŞTIRMA | 42 |
| 3.1 Dönüşüm Modeli | 45 |
| 3.1.1 Fiziksel Modeller | 45 |
| 3.1.2 Radyal Bazlı Fonksiyon Gösterimi- İnce Tabaka Kama Yöntemi . | 51 |

| | | |
|----------|---|------------|
| 3.2 | Benzerlik Ölçümü | 52 |
| 3.2.1 | İmge Fark Ölçümü | 53 |
| 3.2.2 | İmge İlişkisi | 53 |
| 3.2.3 | Karşılıklı Bilgi Miktarı | 53 |
| 3.3 | Düzenlilik Terimi | 54 |
| 3.4 | Optimizasyon | 55 |
| 3.4.1 | Sürekli Optimizasyon | 56 |
| 3.4.2 | Ayrık Optimizasyon | 57 |
| 3.5 | İmge Kalite Değerlendirmesi | 58 |
| 3.5.1 | PSNR | 58 |
| 3.5.2 | SSIM | 58 |
| 3.5.3 | NCC | 59 |
| 3.6 | DeneySEL Sonuçlar | 59 |
| 4 | EVRIŞİMLİ SINIR AĞLARIYLA İMGE KAYNAŞTIRMA | 79 |
| 4.1 | Giriş | 79 |
| 4.2 | Makine Öğrenmesi Sınıfları | 82 |
| 4.2.1 | Denetimli Öğrenme | 82 |
| 4.2.2 | Denetimsiz Öğrenme | 83 |
| 4.2.3 | Yarı Denetimli Öğrenme | 83 |
| 4.2.4 | Pekiştirmeli Öğrenme | 83 |
| 4.3 | Öğrenme Modellerinin Değerlendirilmesi | 83 |
| 4.3.1 | Eğitim, Doğrulama ve Test Veri Setleri | 84 |
| 4.3.2 | Veri Önışleme ve Öznitelik Çıkarma | 85 |
| 4.3.3 | Aşırı Öğrenme ve Eksik Öğrenme | 86 |
| 4.4 | Evrişimli Sinir Ağları | 87 |
| 4.4.1 | Evrişim İşlemi | 87 |
| 4.4.2 | Kenar Etkisi ve Doldurma | 89 |
| 4.4.3 | Evrişim Adım Aralığı | 90 |
| 4.4.4 | En Büyükleri Biriktirme | 90 |
| 4.5 | Otokodlayıcılar | 90 |
| 4.5.1 | Tamamlanmamış Otokodlayıcı | 92 |
| 4.5.2 | Düzenlileştirilmiş Otokodlayıcı | 92 |
| 4.6 | DeneySEL Sonuçlar | 94 |
| 5 | SONUÇ VE ÖNERİLER | 104 |
| | KAYNAKÇA | 106 |
| | TEZDEN ÜRETİLMİŞ YAYINLAR | 113 |

SİMGE LİSTESİ

| | |
|-------------------------------------|---|
| κ | Açısal dalga sayısı |
| \mathbf{w} | Akış vektörü |
| S | Amaç fonksiyonu |
| c | Arama penceresi merkezi |
| δ | Arama yönü |
| I_b | Arka plan modeli |
| λ | Arıtan otokodlayıcı içim hiperparametre |
| c | Artan etiket |
| $\mathcal{P}, \mathcal{T}, \Lambda$ | Basınç, Sıcaklık, Dalga boyu |
| B | Benzeşim metriği |
| η | Bileşen olasılığı |
| θ | Bilinmeyen parametreler |
| d | Bit sayısı |
| φ | Boş uzay |
| p | Boyut sayısı |
| \mathcal{H} | Bozulma modeli |
| ρ | Bulanıklık şiddeti |
| Γ | Ceza payı |
| I_c | Çakıştırılmış imge |
| p | Dağılım fonksiyonu |
| $\mathcal{L}_0, \mathcal{I}_0$ | Dış ölçek, iç ölçek |
| \mathbf{P} | Doğrusal operatör |
| \mathcal{A} | Dürtü gücü |

| | |
|----------------------------|---------------------------------------|
| R | Düzenleştirici terim |
| E | Enerji fonksiyonu |
| w, \hat{w} | Gerçek ve tahmini karışım oranı |
| $\Sigma, \hat{\Sigma}$ | Gerçek ve tahmini kovaryans matrisi |
| $\mu, \hat{\mu}$ | Gerçek ve tahmini ortalama vektör |
| o | Girdi |
| I_{g_s}, I_{f_s} | Gözlenen ve işlenmiş alt imge bölgesi |
| \mathcal{G}, \mathcal{F} | Gözlenen ve işlenmiş model |
| I_m | Hareketli imge |
| $D_v(\mathbf{r})$ | Hız yapı fonksiyonu |
| C_v^2 | Hız yapı parametresi |
| \mathbf{v} | Hız vektörü |
| ζ, α | Hiperparametreler |
| I | İmge |
| m, n | İmge boyutu |
| Z | İmgeler arasındaki fark |
| k | İndis |
| ω, a_0, a_1, a_2 | İTK birincil katsayılar |
| U_r | İTK çözümü |
| z | Karar etiketi |
| $O(\mathbb{L})$ | Karmaşıklık düzeyi |
| \mathbb{I} | Karşılıklı bilgi |
| \mathbf{w} | Katsayı |
| γ | Kazanç faktörü |
| P | Kayıp fonksiyonu |
| W | Kestirme operatörü |
| K, L, M | Kestirme operatörü bileşenleri |
| $D_n(\mathbf{r})$ | Kırılma indisi yapı fonksiyonu |
| C_n^2 | Kırılma indisi yapı parametresi |

| | |
|------------------------|--|
| a | Kod çözücü |
| b | Kodlayıcı |
| Φ_n | Kolmogorov kırılma indisi güç spektrum yoğunluğu |
| q | Komşu pikselin konumu |
| Q | Kontrol noktaları konum matrisi |
| F | Kuvvet |
| χ, ν | Lame'nin elastik sabitleri |
| h | Olasılık |
| v | Ortalama hız |
| α | Öğrenme oranı |
| r | Ölçek büyüklüğü |
| C_1, C_2, C_3 | Parlaklık, kontrast ve yapı bileşenleri |
| x | Piksel konumu |
| I_r | Referans imge |
| r | Seçili iki kontrol noktası arasındaki uzaklık |
| $D_\Theta(\mathbf{r})$ | Sıcaklık yapı fonksiyonu |
| C_Θ^2 | Sıcaklık yapı parametresi |
| H | Shannon entropisi |
| t, d | SIFT imge için eşik değerleri |
| \hat{t} | Sonsal olasılık |
| \hat{y} | Tahmini sınıf üyeliği |
| φ | Takas parametresi |
| N | Toplam piksel sayısı |
| K | Toplam sayı (sınıf, imge) |
| x, y | Uzamsal koordinatlar |
| u, v | Yatay ve dikey frekans değişkenleri |
| u | Yer değiştirme vektörü |
| t | Zaman |

KISALTMA LİSTESİ

| | |
|-------|-------------------------------------|
| CAE | Contractive Autoencoder |
| CC | Correlation Coefficient |
| DAE | Denoising Autoencoder |
| DoG | Difference of Gaussians |
| DTCWT | Dual Tree Complex Wavelet Transform |
| GKM | Gauss Karışım Modeli |
| GPU | Graphics Processing Unit |
| HFD | Hızlı Fourier Dönüşümü |
| İTK | İnce Tabaka Kama |
| JPEG | Joint Photographic Experts Group |
| MTF | Modülasyon Transfer Fonksiyonu |
| NYF | Nokta Yayılım Fonksiyonu |
| NCC | Normalized Cross Correlation |
| OGM | Olasılıksal Grafik Model |
| PSNR | Peak Signal to Noise Ratio |
| PDM | Piksel Doluluğu Modelleme |
| PNG | Portable Network Graphic |
| ROI | Region of Interest |
| SIFT | Scale Invariant Feature Transform |
| SSIM | Structural Similarity Index Measure |
| SSD | Sum of Square Difference |
| TBA | Temel Bileşenler Analizi |
| Th | Threshold |

ŞEKİL LİSTESİ

| | | |
|------------|---|----|
| Şekil 1.1 | Örnek bir imge dizisi ve bu dizi içindeki geometrik deformasyonlar . | 2 |
| Şekil 1.2 | Görüntülenen sahne (sol), türbülanslı ortam (orta), atmosferik türbülans bozulmasına uğramış imge (sağ) | 2 |
| Şekil 1.3 | Çalışmanın İçeriği | 7 |
| Şekil 2.1 | Veri setlerinden temsili imgeler ((a) İmge Seti 1, (b) İmge Seti 2, (c) İmge Seti 3) | 16 |
| Şekil 2.2 | Veri setlerinden temsili imgeler ((a) İmge Seti 4, (b) İmge Seti 5, (c) İmge Seti 6) | 16 |
| Şekil 2.3 | Veri setlerinden temsili imgeler ((a) İmge Seti 7, (b) İmge Seti 8, (c) İmge Seti 9) | 16 |
| Şekil 2.4 | Veri setlerinden temsili imgeler ((a) İmge Seti 10, (b) İmge Seti 11, (c) İmge Seti 12) | 17 |
| Şekil 2.5 | Veri setlerinden temsili imgeler ((a) İmge Seti 13, (b) İmge Seti 14, (c) İmge Seti 15) | 17 |
| Şekil 2.6 | Veri setlerinden temsili imgeler ((a) İmge Seti 16, (b) İmge Seti 17, (c) İmge Seti 18) | 17 |
| Şekil 2.7 | Veri setlerinden temsili imgeler ((a) İmge Seti 19, (b) İmge Seti 20) | 18 |
| Şekil 2.8 | Veri seti-1 için farklı alanlarda yağınlik değişimlerinin belirlenmesi . | 24 |
| Şekil 2.9 | Veri seti-1 Şekil 2.8’de görülen kırmızı çizgi için imge dizisi yağınlik değişimi | 25 |
| Şekil 2.10 | Veri seti-1 Şekil 2.8’de görülen yeşil çizgi için imge dizisi yağınlik değişimi | 25 |
| Şekil 2.11 | Veri seti-1 Şekil 2.8’de görülen mavi çizgi için imge dizisi yağınlik değişimi | 26 |
| Şekil 2.12 | PDM yöntemiyle Veri seti-1 için bozulmaların görsel temsili | 27 |
| Şekil 2.13 | PDM yöntemiyle Veri seti-3 için bozulmaların görsel temsili | 28 |
| Şekil 2.14 | PDM yöntemiyle Veri seti-4 için bozulmaların görsel temsili | 28 |
| Şekil 2.15 | PDM yöntemiyle Veri seti-5 için bozulmaların görsel temsili | 29 |
| Şekil 2.16 | PDM yöntemiyle Veri seti-9 için bozulmaların görsel temsili | 29 |
| Şekil 2.17 | PDM yöntemiyle Veri seti-12 için bozulmaların görsel temsili | 30 |
| Şekil 2.18 | PDM yöntemiyle Veri seti-15 için bozulmaların görsel temsili | 30 |

| | | |
|-------------------|--|----|
| Şekil 2.19 | PDM yöntemiyle Veri seti-16 için bozulmaların görsel temsili | 31 |
| Şekil 2.20 | PDM yöntemiyle Veri seti-17 için bozulmaların görsel temsili | 31 |
| Şekil 2.21 | Veri Seti-1'de (200,200) noktası için GKM gösterimi | 32 |
| Şekil 2.22 | GKM yöntemiyle bozulma bölgelerinin gösterimi | 33 |
| Şekil 2.23 | GKM yöntemiyle Veri seti-1 için bozulmaların görsel temsili | 37 |
| Şekil 2.24 | GKM yöntemiyle Veri seti-3 için bozulmaların görsel temsili | 37 |
| Şekil 2.25 | GKM yöntemiyle Veri seti-4 için bozulmaların görsel temsili | 38 |
| Şekil 2.26 | GKM yöntemiyle Veri seti-5 için bozulmaların görsel temsili | 38 |
| Şekil 2.27 | GKM yöntemiyle Veri seti-6 için bozulmaların görsel temsili | 39 |
| Şekil 2.28 | GKM yöntemiyle Veri seti-12 için bozulmaların görsel temsili | 39 |
| Şekil 2.29 | GKM yöntemiyle Veri seti-15 için bozulmaların görsel temsili | 40 |
| Şekil 2.30 | GKM yöntemiyle Veri seti-16 için bozulmaların görsel temsili | 40 |
| Şekil 2.31 | GKM yöntemiyle Veri seti-17 için bozulmaların görsel temsili | 41 |
| Şekil 3.1 | SIFT-Akış yönteminde kabadan inceye eşleştirme algoritması [60] . . | 50 |
| Şekil 3.2 | Veri seti-1 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 63 |
| Şekil 3.3 | Veri seti-2 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 64 |
| Şekil 3.4 | Veri seti-3 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 65 |
| Şekil 3.5 | Veri seti-4 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 66 |
| Şekil 3.6 | Veri seti-7 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 67 |

| | | |
|-------------------|---|----|
| Şekil 3.7 | Veri seti-8 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 68 |
| Şekil 3.8 | Veri seti-9 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 69 |
| Şekil 3.9 | Veri seti-13 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 70 |
| Şekil 3.10 | Veri seti-14 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 71 |
| Şekil 3.11 | Veri seti-19 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 72 |
| Şekil 3.12 | Veri seti-20 için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 73 |
| Şekil 3.13 | Door [3] için karşılaştırma sonuçları: <u>1.satır</u> : (sol) Bozulmuş İmge, (sağ) Referans İmge, <u>2.-4. satırlar için</u> : (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, <u>2.satır</u> : Demons Yöntemi, <u>3.satır</u> : İTK Yöntemi, <u>4.satır</u> : SIFT-Akış Yöntemi | 74 |
| Şekil 3.14 | Tepe sinyali gürültü oranı (PSNR) metrik sonuçları | 75 |
| Şekil 3.15 | Yapısal benzerlik indeksi ölçümü (SSIM) metrik sonuçları | 76 |
| Şekil 3.16 | İlinti katsayısı (NCC) metrik sonuçları | 76 |
| Şekil 3.17 | Çakıştırma yöntemlerinin işlem süreleri | 77 |

| | |
|--|-----|
| Şekil 3.18 Veri seti-1 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra | 77 |
| Şekil 3.19 Veri seti-3 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra | 78 |
| Şekil 3.20 Veri seti-4 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra | 78 |
| Şekil 3.21 Veri seti-8 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra | 78 |
| Şekil 4.1 Derin ağ örneği | 80 |
| Şekil 4.2 Ağ, katmanlar, kayıp işlevi ve optimizasyon arasındaki ilişki | 81 |
| Şekil 4.3 K-fold doğrulama | 85 |
| Şekil 4.4 Örnek bir evrişimli sinir ağı modeli | 88 |
| Şekil 4.5 Örnek sinir ağı modelinin yapısı | 88 |
| Şekil 4.6 Otokodlayıcının genel yapısı | 91 |
| Şekil 4.7 Örnek bir otokodlayıcı modeli | 91 |
| Şekil 4.8 Arıtan otokodlayıcının genel yapısı | 94 |
| Şekil 4.9 Önerilen kodlayıcı mimarisi | 95 |
| Şekil 4.10 Önerilen kod çözücü mimarisi | 95 |
| Şekil 4.11 Veri seti-1 için imge kaynaştırma sonuçları | 98 |
| Şekil 4.12 Veri seti-1 için örnek bir alt imge sonucu | 98 |
| Şekil 4.13 Veri seti-2 için imge kaynaştırma sonuçları | 99 |
| Şekil 4.14 Veri seti-2 için örnek bir alt imge sonucu | 99 |
| Şekil 4.15 Veri seti-4 için imge kaynaştırma sonuçları | 100 |
| Şekil 4.16 Veri seti-4 için örnek bir alt imge sonucu | 100 |
| Şekil 4.17 Veri seti-15 için imge kaynaştırma sonuçları | 101 |
| Şekil 4.18 Veri seti-15 için örnek bir alt imge sonucu | 101 |
| Şekil 4.19 Veri seti-15 için imge kaynaştırma sonuçları | 102 |
| Şekil 4.20 Veri seti-17 için örnek bir alt imge sonucu | 102 |
| Şekil 4.21 Veri seti-door [3] için imge kaynaştırma sonuçları | 103 |
| Şekil 4.22 Veri seti-door [3] için örnek bir alt imge sonucu | 103 |

TABLO LİSTESİ

| | | |
|------------------|--|----|
| Tablo 3.1 | Demon's Yöntemi için Kalite Metrik Sonuçları ve İşlem Süresi | 60 |
| Tablo 3.2 | İTK Yöntemi için Kalite Metrik Sonuçları ve İşlem Süresi | 61 |
| Tablo 3.3 | SIFT-Akış Yöntemi için Kalite Metrik Sonuçları ve İşlem Süresi | 61 |
| Tablo 4.1 | Yapısal Benzerlik Ölçümü Metriği (SSIM) Sonuçları | 96 |
| Tablo 4.2 | Tepe Sinyal Gürültü Oranı (PSNR) Metriği Sonuçları | 97 |
| Tablo 4.3 | İlinti Katsayısı (NCC) Metriği Sonuçları | 97 |

Uzak Mesafeli Görüntülemelerde Bozulmaların İncelenmesi ve İyileştirilmesi

Gülsüm GEZER

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Doktora Tezi

Bu çalışmada, uzak mesafeli görüntülemelerde elde edilen imge setleri üzerinde oluşan atmosferik türbülans orijinli bozulmaların tespiti ve şiddetinin temsili ile orijinal sahnenin kestirimi için bozulmaları azaltmaya yönelik bozulmuş imgelerin kaynaştırılması iki temel hedeftir.

Bu amaçla öncelikle 300 m ila 3.5 km aralığında değişen mesafelerde 20 sahnenin imge dizisi elde edilmiştir. Böylece, elde edilen imge setleri değişen seviyelerde atmosferik türbülans etkilenmiştir.

Atmosferik türbülans bozulmaları, uzamsal/zamansal olarak dalgalanan bozulma türleridir. Bu nedenle deneysel veya gözlemsel model doğrudan türetilemez. Elde edilecek matematiksel model için; sıcaklık, nem ve basınca göre değişen küçük dalgalanmalar şeklindeki havanın kırılma indisinin niceliksel ölçümlerinin o sırada elde edilmiş olması gerekir. Öte yandan, bozulmuş imge setlerinin kapsamlı bir şekilde anlaşılmasını içeren çalışmalar literatürde daha yaygındır. Bu çalışmada, bozulmaların tespiti ve şiddetinin temsili için genel olarak Gauss Karışım Modelleri ve Piksel Doluluğu Modelleri üzerinde durulmuştur.

İmge dizileri üzerinde atmosferik türbülans bozulmalarının oluşturduğu geometrik deformasyonların minimize edilmesinde katı olmayan imge çakıştırma yöntemlerinin etkinliği farklı dönüşüm modelleri (fiziksel ve radyal tabanlı modeller) göz önünde bulundurularak ele alınmış ve kullanılan yöntemlerin performansı nesnel kalite metrikleri dikkate alınarak değerlendirilmiştir.

Kalite metrikleri sonucuna göre en etkin imge çakıştırma yönteminden elde edilen

akıřtırılmıř imge dizilerinden orijinal sahneye yakın bir imgenin kestirimi iin imge kaynařtırma yntemi olarak evriřimli sinir ađlarından biri olan arıtan otokodlayıcı modeli geliřtirilmiř ve bu modelin etkinliđi literatrde bulunan bazı geleneksel imge kaynařtırma teknikleriyle hem grsel hem de kalite metrikleri kullanılarak karřılařtırılmıřtır.

Anahtar Kelimeler: Atmosferik trblans, İmge akıřtırma, İmge kaynařtırma, Gauss karıřım modeli, SIFT-Akıř yntemi, Evriřimli sinir ađları



Analysis and Restoration of Degradations in Long Range Imaging

Gülsüm GEZER

Department of Control and Automation Engineering

Doctor of Philosophy Thesis

In this study, two main objectives are the detection and quantitative representation of atmospheric turbulence originated distortions imposed on image sets acquired in long-range imaging, as well as the fusion of the degraded images to reduce the distortions on scene estimations.

For this purpose, we put our efforts firstly towards long-range image acquisition; multiple images of 20 scenes at varying distances in the range of 300 m to 3.5 km are obtained. Therefore, the acquired images are affected by atmospheric turbulence with varying levels.

Atmospheric turbulence distortions are spatially/temporally wavering types of distortions, and the experimental or observational model cannot be derived straightforwardly. For the mathematical model to be obtained, quantitative measurements of the refractive index of air in the form of small fluctuations varying with temperature, humidity, and pressure have to be handy at any time instance during the acquisition. On the other hand, the efforts involving the comprehensive understanding of the degraded image sets are more in common in the literature. In this study, we focused mainly on Gaussian Mixture Models and Pixel Occupancy Models for the detection and quantitative representation of the distortions.

The aptness of the state of art non-rigid image registration methods associated with certain transformation model types (physical and radial basis models) in minimizing the geometric deformations of the images originated essentially due to atmospheric turbulence effects is discussed. The performance of the utilized methods is evaluated considering objective quality metrics.

The best-aligned image sequences of each dataset - considering the results of the quality metrics- are introduced to the image fusion stage. The main contribution of this dissertation can be regarded as the employment of a convolutional neural network model, denoising autoencoder for the fusion of the images sequences towards estimation of the original scene freed from atmospheric turbulence effects. The effectiveness of the model is revealed both visually and quantitatively through experimental studies involving some other well-known conventional fusion methods.

Keywords: Atmospheric turbulence, Image registration, Image fusion, Gaussian mixture model, SIFT-flow registration, Convolutional neural network



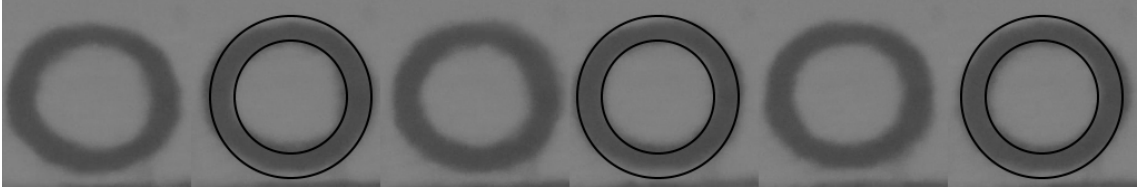
1.1 Literatür Özeti

Görüntüleme; mevcut sahnenin, tamamının veya bir bölümünün imge ya da video çekme suretiyle kayıt altına alınmasına denir. İmge ise, bu sahnenin bir görüntüleme cihazıyla elde edilmiş iki boyutlu (2B) halidir. İmgeye üç boyutlu uzayda herhangi bir nesnenin veya sahnenin iki boyutlu haritası da denebilir ve böylece bir $I(x, y)$ fonksiyonu olarak tanımlanabilir. x ve y uzamsal koordinatlar (konum bilgisi), I fonksiyonun (x, y) noktasındaki genliği de yeğinlik (intensity) veya gri seviyesi olarak adlandırılır. Burada imgenin herhangi bir (x, y) konumundaki her bir elemanı piksel (resim elemanı) olarak adlandırılır ve her pikselin sahip olduğu bir yeğinlik değeri vardır. I fonksiyonu içindeki bütün noktalarda bu değerler sonlu ve ayrık büyüklükte olduğunda, imge sayısallaştırılmış olduğundan bu tip imgelere sayısal imge denir. İmgedeki detayları belirleyen en önemli faktörler, imgenin boyutları ve piksel yeğinliğidir.

Normal şartlar altında imgede, ilgilenilen sahneye gürültü ve bulanıklık eşlik eder. Bu durumun nedenleri sensörden kaynaklanan gürültüler, cihazın kendisinin çevresel titreşimlere maruz kalması, o anki sahnede bulunan hareket eden objelerden dolayı oluşan hareket bulanıklığı, odak dışı bulanıklık veya çevresel şartlar vb. olabilir.

Uzak mesafeli görüntülemelerde havanın sıcaklığı, basınç, rüzgarın şiddeti vb. doğa olaylarının neden olduğu atmosferik türbülans, imgelerde yerel dalgalanmalara ve geometrik deformasyonlara neden olmaktadır. Şekil 1.1'de 3.5 km uzaklıktan elde edilmiş bir imge dizisinde atmosferik türbülans kaynaklanan geometrik deformasyonlar görülmektedir.

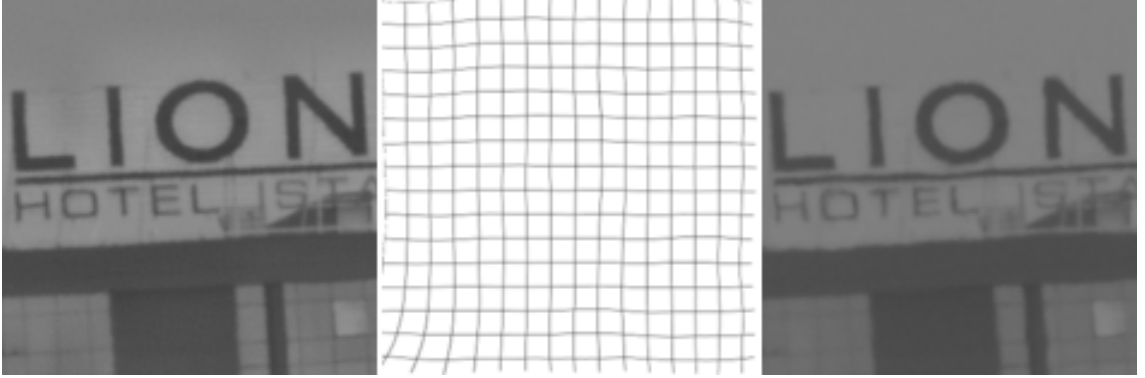
Atmosferik türbülans kaynaklanan imge bozulmalarında, hareket bulanıklığı veya odak dışı bulanıklığı gibi tekdüze bir bozulmadan bahsedilemez. Bu nedenle imge iyileştirme işleminde imge, ya alt parçalara bölünüp tekdüze bulanıklık giderici filtreler kullanılır [1] ya da imgenin -aynı sahne olmak koşuluyla- farklı zamanlarda



Şekil 1.1 Örnek bir imge dizisi ve bu dizi içindeki geometrik deformasyonlar

birden fazla temsili kullanılması gerekir [2], [3], [4], [5], [6], [7]. Duruk imgeler için aynı sahnenin çoklu temsili daha fazla bilgi bulunmaktadır.

Şekil 1.2’de sol kısımda bozulmamış bir imge, orta kısımda atmosferik türbülanslı ortamın benzetimi ve sağ kısımda da bu ortamdan elde edilmiş bozulmuş imge görülmektedir.



Şekil 1.2 Görüntülenen sahne (sol), türbülanslı ortam (orta), atmosferik türbülans bozulmasına uğramış imge (sağ)

İmge iyileştirmesi, bozulmuş ve/veya gürültülü bir imgenin orijinal haline yakın bir kestirimini elde etmeye yarayan işlemlerin bütünüdür. Medikal uygulamalardan (manyetik rezonans, bilgisayarlı tomografi, akciğer bölütleme vb.), uzay araştırmalarına (örneğin yıldız, asteroid, galaksi vb. elde edilen imgelerin tanımlanması), güvenlikten (yüz tanıma vb.), ulaşıma (plaka tanıma vb.) pek çok alanda ihtiyaç duyulan imge iyileştirme işlemi son yıllarda araştırmacıların üzerinde çalıştığı önemli konulardan biri haline gelmiştir. Gelişen teknolojiyle birlikte, imgede oluşan bu tarz bozulmaların etkili bir şekilde giderilmesi önem arz etmektedir.

İmge iyileştirme problemlerinde genelde bozulma işleminin Nokta Yayılım Fonksiyonu (NYF) adında, lineer zamanla değişmeyen bulanıklık modeli içerdiği kabul edilir. Eğer bozulma modeli elde edilemiyorsa yani NYF bilinmiyorsa, bu problemin çözümü gözü kapalı ters evrişim (blind deconvolution) olarak bilinir. Literatürde, bu problem ile ilgili başarılı çözümler bulunmaktadır [8], [9], [10]. Bu yöntemlerin genel kabulü, bütün imgenin tek bir NYF tarafından bozulduğu görüşüdür. Fakat bu, bozulmuş her imge için etkili bir model değildir.

Atmosferik türbülans etkileriyle başa çıkmak için iki tür yaklaşım benimsenebilir. Bunlardan ilki atmosferin bulanıklık düzeyini modelleyecek bir analitik ifade bulmaktır. Atmosferik türbülansın NYF'si hakkında bir ön bilgi yoktur. Türbülansın uzun vadeli etkisini modellemek için [11] çalışmasında Denklem 1.1'deki optik transfer fonksiyonu (NYF'nin Fourier dönüşümü) türetilmiştir:

$$\mathcal{H}(u, v) = e^{-\rho(u^2+v^2)^{5/6}}, \quad (1.1)$$

Burada \mathcal{H} ; bozulma modelini, (u, v) ; yatay ve dikey frekans değişkenlerini, ρ ise bulanıklığın şiddetini temsil etmektedir. ρ arttıkça NYF'nin büyüklüğü de artmaktadır. Bu modelleme ile \mathcal{H} 'nin kestirimi, ρ parametresinin belirlenmesine indirgenir. ρ 'nun tahmini, [12] ve [13] çalışmalarında basıklığın minimize edilmesi yaklaşımıyla elde edilmektedir. Basıklık burada, iyileştirilmiş imgelerin kalitesi için bir ölçüt olarak kullanılmıştır. Atmosferin bulanıklık düzeyini belirlemek adına yapılan başka bir çalışmada ise David Fried [14] atmosferin modülasyon transfer fonksiyonunu (MTF) modellemek için, optik çözünürlük limitini tanımlamada kullanılan Fried parametresini (atmosferik etkilerle oluşan optik sapmaların ortalama karekökünün 1 radyana eşit olduğu dairesel alanın çapı) önermiştir. [15] ve [16]'da, Fried çekirdeği (fried kernel- iki terimden oluşur. Türbülansın önemsiz olduğu durumlarda atmosferin ve sistemin toplam MTF'si ve türbülans kaynaklı faz eğiminin etkisini modelleyen MTF) ile birlikte bir gözü kapalı ters evrişim yöntemi, atmosferik türbülans bulanıklığıyla başa çıkmak için önerilmiştir.

[17]'de araştırmacılar, McGlamery'nin [18] çalışmasında belirttiği hızlı Fourier dönüşümü tabanlı yöntemini kullanıp bağımsız faz ekranları oluşturarak bir türbülans benzetimi gerçekleştirmişlerdir. [19], [20], [21] ve [22]'de ise atmosferik türbülansın imge kalitesi üzerindeki etkilerinin benzetimi için bilgisayar modeli önerilmiştir.

Atmosferik türbülans etkileriyle başa çıkmak için ikinci yaklaşım ise imge dizilerine önce imge çakıştırma ve ardından imge kaynaştırma işleminin yapılmasıdır. İmge çakıştırma; mevcut bozulmuş imgelerin belirli bir referans imgeye göre karşılıklı noktaları denk gelecek şekilde çakıştırılması işlemidir. Burada referans imge, imge dizilerinin zamansal ortalamalı bir şekli olabileceği gibi imge dizisi içinden seçilmiş herhangi bir imge de olabilir. [23]'te uzamsal/zamansal ağırlıklı bir ortalamaya dayanan basit ve genel bir referans imge oluşturma tekniği tanıtılmıştır.

İmge çakıştırma kısaca imgelerin aynı uzamsal koordinat sistemi üzerinde hizalanması işlemidir. Atmosferik türbülans etkilenen imgelerde nesnelere serbest formda şekil değişikliğine uğradığı için etkili bir dönüşüm modeli önem kazanmaktadır.

Bu konuda yapılan çalışmalar şöyle sıralanabilir. Yapılan en eski çalışmalar Benek İmgeleme (Speckle Imaging) [24] adı altında uzay arařtırmalarında elde edilen imgelerin görüntü kalitesinin düzeltilmesine yöneliktir. Carrano da [25]'te benek imgeleme tekniğini kullanmıştır. Benek imgelemenin amacı, aynı sahnenin imge dizilerinden kırım-sınırlı bir kestirim elde etmektir. Bu da, imgenin ayrı ayrı Fourier dönüşümü genliğini ve Fourier dönüşümü fazını kestirip, ardından ters fourier dönüşümü uygulamakla yapılır. [6], [26] ve [27] çalışmalarında arařtırmacılar tarafından atmosferik türbülans için öncelikle imge dizilerinde, optikler üzerinde tozla kaplanan kötü ve işe yaramayan piksellerin yerleri haritalanıp yerlerine etraflarındaki aktif piksellerin ortalamaları konulmuş, ardından Fourier dönüşümü tabanlı imge çakıştırma yapılmıştır. Yapılan başka bir etkili çalışmada [28] ise çakıştırma işlemi belirlenen bir referans ve imge dizisi arasında bozulmuş imgedeki her bir pikselin referans imgede karşılık gelen piksele göre yerdeğıştirmesinin bir fonksiyonu olarak benzerliğinin ölçüldüğü çapraz korelasyon yöntemiyle sağlanmıştır.

Şanslı bölge tekniğinde, imge dizilerinde her bir imge için hesaplanan nesnel kalite metrikleriyle karşılaştırılıp en kaliteli bölgelerin seçimi yapılarak bu bölgeler kaynaştırılır ve imgedeki yerel dalgalanmaların giderilmesi için etkili yöntemlerden biridir. [29]'da şanslı bölgelerin tespiti ikiz uyumluluk analiziyle tespit edilmiştir. Video görüntülerinde bulunan her bir imgeyi keskinleştirmek için [30]'da Sobolev gradyan akış yöntemi ve zamansal değışen bozulmaları azaltmak için de laplace operatörü birlikte kullanılmıştır. Yine başka bir çalışmada [31] bozulmalar, imge dizilerine sırasıyla uzamsal dönüşüm, bilinear ara değıerleme, adaptif yörünge tahmini ve kayma vektörleri kullanılarak imgede hesaplanan noktaların yer değıştirmesi işlemleri uygulandıktan sonra çakıştırılmış imge dizileri elde edilmiştir.

Atmosferik türbülans bozulmalarında imge çakıştırma algoritmalarının performanslarının karşılaştırıldığı çalışmalar da mevcuttur. [32]'de, Fourier dönüşümü tabanlı imge çakıştırması, İnce tabaka kama (thin plate spline- İTK) ve Demons yöntemleri karşılaştırılmıştır. [33] çalışmasında yapay atmosferik türbülans oluşturulan ortamdaki elde edilen imgelerde iki farklı çakıştırma algoritması olan blok eşleme ve wiener filtresi yaklaşımları uygulanmış, [34]'te ise bu yöntemlerin performansları karşılaştırılmıştır. Arařtırmacılar bu çalışmalarını yaparken uzamsal değışen bulanıklığa sahip bir imge dizisinin benzetimi için bir Monte-Carlo dalga optiğı yaklaşımını kullanmışlardır. Karşılaştırma yaptıkları yöntemler ise şanslı bölge tekniğı, ikili benek imgeleme ve blok eşleştirme yaklaşımlarıdır.

Çakıştırılmış imge dizilerinin bir kaynaştırma metoduyla birleştirilip geometrik deformasyonları azaltılmış tek bir imgeye dönüştürülmesi için etkin bir yöntem ihtiyacı vardır. Örneğin farklı sensörlerden elde edilmiş imgelerin kaynaştırılmasında

kullanılan teknikler, medikal imgelerin kaynaştırılmasında işe yaramayabilir. Aynı durum atmosferik türbülans bozulmasına sahip imgeler için de geçerlidir.

Kaynaştırma terimi genel olarak birkaç alanda elde edilen önemli bilgilerin çıkarılmasına dayanan bir yaklaşımdır. İmge kaynaştırmanın amacı çok sensörlü, çok zamanlı veya çok görüşlü yaklaşımlarında bulunan anlamlı bilgilerin birleştirilerek tek bir imgeye toplanmasıdır.

İmge kaynaştırma birçok uygulama alanında kullanılır. Uzaktan algılama ve astronomide çok sensörlü kaynaştırma biri yüksek uzamsal çözünürlüğe, diğeri yüksek spektral çözünürlüğe sahip iki sensörden gelen imgeleri birleştirerek yüksek uzamsal ve spektral çözünürlük elde etmek için kullanılır. Medikal görüntüleme bilgisayarlı tomografi, manyetik rezonans görüntüleme ve pozitron emisyon tomografi imgelerinin eş zamanlı değerlendirilmesi gibi çok sayıda kaynaştırma uygulaması mevcuttur. Görünür ve kızılötesi imgelerin kaynaştırılmasına dayanan askeri, güvenlik ve gözetim alanlarında çok sayıda uygulama vardır. Çok görüşlü imge kaynaştırma durumunda, aynı sensör tarafından aynı sahnenin farklı görüş açılarından alınan imge dizisinin, sensörün sağladığından daha yüksek çözünürlüklü bir imge elde etmek veya üç boyutlu temsili oluşturmak için birleştirilir. Çok zamanlı imge kaynaştırma ise iki farklı amaç taşır. Aynı sahnenin imge dizileri, sahnedeki değişiklikleri bulmak ve değerlendirmek veya aynı sahnenin daha az bozulmuş bir halini elde etmek için farklı zamanlarda toplanır. İlk amaç, medikal görüntüleme tümörlerin veya organların değişiminin tespiti için veya uzaktan algılama alanında arazi izlemede kullanılır. İkinci amaç ise, örneğin uzak mesafe görüntüleme imge dizilerinde oluşan atmosferik türbülans bozulmalarını azaltmakta kullanılır.

Tüm imge kaynaştırma görevlerine uygulanabilen evrensel bir yöntem tasarlamak imkansızdır. Her imge kaynaştırma yönteminde sadece kaynaştırma amacı ve her bir sensörün özelliği değil, aynı zamanda belirli görüntüleme koşulları, görüntüleme geometrisi, gürültü, gereken doğruluk ve uygulamaya bağlı veri özellikleri de hesaba katılmalıdır [35].

Süper çözünürlük konusu, uzak mesafe görüntüleme imge iyileştirme adımı için önerilen çalışmalardan biridir. Süper çözünürlüklü imge oluşturma işlemi, imge dizilerinin hizalanması ve süper çözünürlüklü imgenin yeniden oluşturulması olmak üzere iki temel adımdan oluşur. İmgelerin hizalanması süper çözünürlüklü imge oluşturma işleminin en önemli parçasıdır. İmgelerin hizalanması işleminde iki boyuttaki hareket parametrelerinin (öteleme ve dönme) tahminleri yapılır. [36] ve [37]'de süper çözünürlük yaklaşımı atmosferik türbülans bozulmasına sahip duruk imge dizilerine uygulanmıştır. [38] çalışmasında ise, atmosferik türbülans etkilerinin

orta seviyede olduđu bir ortamda dinamik imgeler için, dinamik süper çözünürlük uygulanarak bir türbülans azaltma yoluna gidilmiştir.

Yapılan etkili çalışmalardan biri de Dalong Li tarafından temel bileşenler analizi kullanılarak önerilmiştir [13]. [39] çalışmasında, iyileştirme yöntemi olarak tekil değer ayrışımı tekniği önerilmektedir. Burada iki boyutlu bozulma modeli, yatay ve dikey yönlerde tek boyutlu iki bozulma modeline ayrıştırılmaktadır.

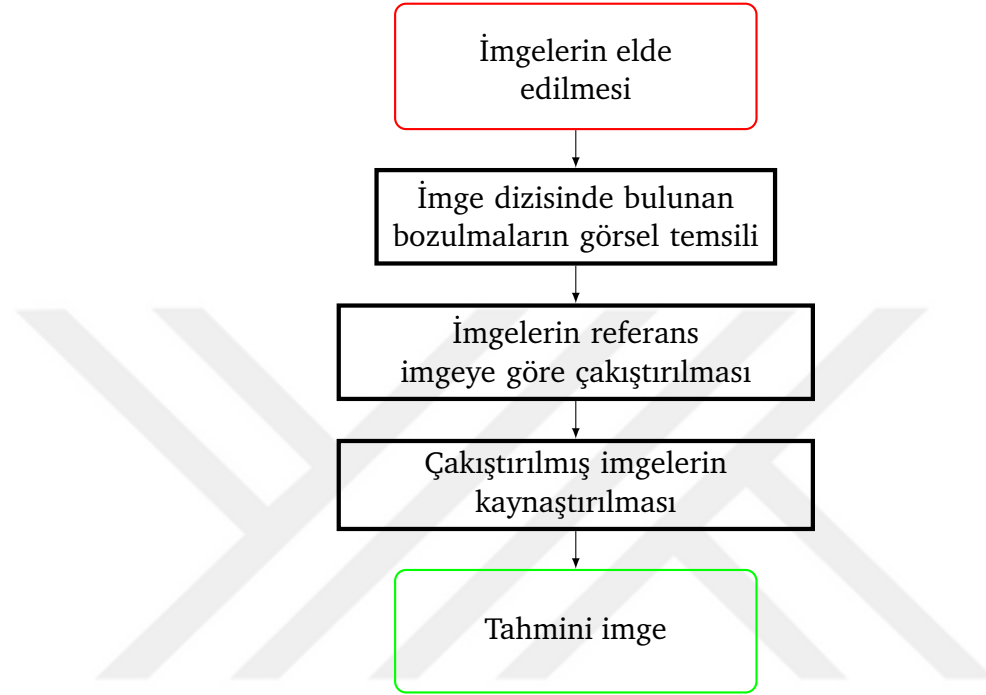
Anantrasirichai [4] ve [5] çalışmalarında uzamsal değişen bozulmalar için katı olmayan bir imge çakıştırma metoduyla birlikte, çift ağaç karmaşık dalgacık dönüşümü (dual tree complex wavelet transform) tabanlı bir imge kaynaştırma metodu önermiştir.

Atmosferik türbülans kaynaklı bozulmaya uğramış imgelerden matematiksel işlemlerle orjinal sahneye yakın bir kestirim elde edilmek isteniyorsa, imge dizilerine ihtiyaç vardır. Bu imgelerde orjinal sahnede bulunan bütün ayrıntılar her bir imgede gelişigüzel bulunacağından öncelikle yapılması gereken işlem, etkin bir imge çakıştırma yöntemi kullanmak olacaktır. İmgelerde yerel dalgalanmalar geometrik deformasyon olarak görüneceğinden, sahnedeki nesnenin şekil değişikliğini hizalamak için en basit dönüşüm modeli olan katı bir imge çakıştırma modeli işimize yaramayacaktır. Çünkü katı dönüşüm, sahnelerin dönmesine ve yer değiştirmesine izin verir ve hatta afin dönüşümle (affine transformation) sahneyi ölçekleyip kırabiliriz. Fakat bu dönüşüm şekil değişikliğini gidermekte işe yaramayacaktır. Böylece daha serbest bir form olan katı olmayan dönüşüm modellerinden birine ihtiyaç duyulacaktır. Burada önemli olan konu, dönüşüm modeline ve optimizasyon kriterlerine karar vermektir. Bu nedenle literatürde bulunan etkin katı olmayan imge çakıştırma yöntemlerinin işlem sonuçlarını nesnel kalite metrikleriyle karşılaştırmak bu konuda iyi bir fikir verebilir.

Son 20 yıl içerisinde makine öğrenmesi alanında hem donanımsal (bilgisayarlarda bulunan grafik işlemci birimi-GPU), hem yazılımsal (yeni optimizasyon algoritmalarının ortaya çıkarılması), hem de veri seti (Youtube, Instagram, Facebook vb. platformlarda bulunan veriler) konusundaki gelişmelerden dolayı evrimsel sinir ağları, imge işleme uygulamalarında özellikle imge sınıflandırma, nesne tanıma v.b. konularında çok iyi sonuçlar ortaya çıkarmaktadır. Bu nedenle çakıştırılmış imgelerin tek bir imgeye kaynaştırılmasında evrimsel sinir ağları yöntemi önerilmektedir.

Şekil 1.3'te bu çalışmanın akış diyagramı görülmektedir. Bu akış diyagramında birinci aşamada atmosferik türbülans etkisiyle bozulmuş imgeler, doğal ortamlarında farklı mesafelerden (> 100 metre) toplanmaktadır. İkinci aşamada, elde edilen bu imgelerde bulunan yerel dalgalanmalar ve bulanıklık iki çeşit modelleme

yöntemiyle görsel olarak temsil edilmektedir. Üçüncü aşamada bozulmaya uğramış yerel bölgelerde düzeltme yapılabilmesi için ilk adım olan katı olmayan imge çakıştırma yöntemleri kullanılarak etkileri ortaya konmaktadır. Son olarak dördüncü aşamada, çakıştırılmış imge dizilerinin geometrik deformasyonu azaltılmış tek bir imgeye kaynaştırılması için önerilen kaynaştırma yöntemi uygulanarak, sonuçları sunulmaktadır.



Şekil 1.3 Çalışmanın İçeriği

1.2 Tezin Amacı

Uzak mesafeden (>100 metre) elde edilen imgelerde atmosferik türbülans orijinli oluşan bozulmalar uzamsal değişen bulanıklık oluşturacağından gözlemsel veya deneysel olarak modellenememektedir. Bozulmanın matematiksel modelinin çıkarılabilmesi için de havanın kırılma indisi yapı sabiti, sıcaklık, basınç vb. parametrelerin ölçülmesi gerekmektedir. Bu parametrelerin bazılarının ölçülmesi pahalı donanımlar gerektirdiğinden imge dizisinde oluşan yerel dalgalanmalar ve geometrik deformasyonların bulunduğu bölgelerin tespitini ve bozulmanın şiddetini, farklı veri modelleme yöntemleriyle görsel olarak temsil etmektir.

Uzak mesafeli görüntüleme konusunda çalışan araştırmacılar için veri seti elde etmek zordur ve bu alanda kullanılacak gerçek türbülans etkisi bulunan imge setlerine erişim kısıtlıdır. Bu nedenle bu tezin önemli bir amacı da bu çalışma için toplanan veri setlerinin, bu alanda çalışmak isteyen araştırmacıların kullanımına sunulacak olmasıdır.

İmgede yerel dalgalanmalar geometrik deformasyon olarak görüldüğü için bu bozulma türünün azaltılmasına yönelik çalışmalar bize katı imge çakıştırma yöntemlerinin bu senaryoda etkili olmadığını göstermektedir. Bu nedenle bozulmuş imge dizileri üzerine katı olmayan imge çakıştırma yöntemlerinden farklı algoritmalar uygulamanın amacı farklı dönüşüm modellerinin etkilerini gözlemlemek ve nesnel ölçüm metrikleriyle performanslarını karşılaştırmaktır.

Önerilen imge kaynaştırma yönteminin etkinliğini ölçmek adına literatürde bulunan bazı geleneksel yöntemlerle birlikte kaynaştırılmış imgelerin görsel ve nesnel ölçüm sonuçlarını sunmaktır.

1.3 Orjinal Katkı

Bu çalışmanın literatüre katkısını şu şekilde sıralayabiliriz:

- Bu alanda var olan bir açık kaynak veri seti eksikliği mevcuttur. Bu nedenle çalışmamızın ilk basamağını uzak mesafe olarak kabul edebileceğimiz (100m'den daha uzak) farklı mesafelerden toplanmış ve farklı bozulma şiddetine sahip veri setleri oluşturmaktadır.
- İmge çakıştırması sonrası elde edilen çakıştırılmış imge dizilerinin, geometrik deformasyonu azaltılmış tek bir imgeye dönüştürülmesi için geliştirilen imge kaynaştırma yöntemi olarak evrişimli sinir ağlarından biri olan arıtan otokodlayıcı modeli uygulanmış ve sonuçları literatürde bulunan bazı geleneksel imge kaynaştırma yöntemleriyle hem görsel olarak hem de nesnel ölçüm metrikleriyle karşılaştırılmıştır.

1.4 Motivasyon

Birçok uygulamada kullanılan uzak mesafeden elde edilmiş imgelerde oluşan atmosferik türbülans kaynaklı bozulmalar, sahne üzerinde yerel dalgalanmalar ve geometrik deformasyonlar şeklinde gözlenmektedir. Bu bozulma türü modellemesi zor olan bir konu olduğu için literatürde genelde bozulmayı modellemek yerine azaltmak üzerine çalışmalar mevcuttur. Atmosferik türbülans bozulma modelini matematiksel olarak elde etmek için gereken bazı değişkenlerin ölçümlerinin sağlanamaması durumunda elde edilmiş ham imge dizilerine ön plân çıkarımı yöntemleri uygulanarak bozulmaların imgenin hangi bölgelerinde yoğunlaştığına ve bozulma şiddetine yönelik görsel bir temsil sunulabilir.

Katı olmayan imge çakıştırma yöntemlerinde farklı dönüşüm modellerinin (fiziksel modeller ve radyal tabanlı gösterim modeli) atmosferik türbülans

dolayı bozulmaya uğramış imgeler üzerindeki performanslarının nesnel kalite metrikleriyle karşılaştırılması hangi algoritmaların daha uygun olduğuna dair bir fikir oluşturacaktır.

Literatürde çoklu odaklı imge kaynaştırma işlemlerinde ve termal-görünür kameralardan elde edilen imgelerin kaynaştırılmasında evrişimli sinir ağlarının kullanıldığı az sayıda çalışma mevcuttur fakat bu yöntemle türbülans etkili imgelerin kaynaştırılmasında yapılan çalışmalar yok denecek kadar azdır. Çakıştırılmış imgelerin orjinal sahneye yakın bir kestirimi için yapılan imge kaynaştırma işleminde nesnel kalite metrikleri kullanılarak, önerilen evrişimli sinir ağları tekniği ile literatürde bulunan bazı imge kaynaştırma yöntemlerinin performansları karşılaştırılarak yöntemin etkinliği ortaya çıkarılabilir.

1.5 Tezin İçeriği

Bölüm 2’de, bütün imge setlerinin elde edilmesi işlemleri ve bu setlerle ilgili bilgiler verilerek, bu imge setlerinden seçilen bazı setlerde oluşan yerel dalgalanmalar ve geometrik deformasyonların bulunduğu bölgelerin tespiti ve bozulmanın şiddeti PDM ve GKM yöntemleriyle görsel olarak temsil edilmektedir.

Bölüm 3’te, bozulmuş imgelerin iyileştirilmesi için ilk basamak olan imge çakıştırma konusunda katı olmayan çakıştırma yöntemlerinden Demons, İTK ve SIFT-Akış yöntemleri uygulanmakta ve performansları nesnel kalite metrikleriyle karşılaştırılmaktadır.

Bölüm 4’te, imge iyileştirmesinin ikinci basamağı olan imge kaynaştırma tekniği olarak derin öğrenme konularından biri olan evrişimli sinir ağları tabanlı bir yöntem önerilmekte ve sonuçları literatürde bulunan diğer bazı imge kaynaştırma yöntemleriyle karşılaştırılmaktadır.

Bölüm 5’te, yapılan çalışmalar kısaca özetlenmekte olup, deneysel sonuçlar yorumlarken ileriki zamanlarda yapılabilecek çalışmalar için öneriler sunulmaktadır.

UZAK MESAFELİ GÖRÜNTÜLEMEDE İMGELERİN ELDE EDİLMESİ VE OLUŞAN BOZULMALAR

2.1 Giriş

Çevresel şartlar içerisinde bulunan hava sıcaklığı, rüzgar şiddeti, ortam basıncı v.b durumlarda gözlenen normal dışı değerler, toplanan imgelerde atmosferik türbülans etkilerinin görülmesine neden olur. Bu şekilde oluşan bozulmalar da imgelerde geometrik deformasyon ve bulanıklık olarak gözlemlenir.

Yer tabakası ve hava arasında sıcaklık farkı oluştuğunda ortam ısınmaya başlar ve yatay hava tabakaları oluşur (laminar akış). Eğer bu sıcaklık farkı artarsa, her bir tabakanın kalınlığı azalır ve bu tabakalar hızlıca hareket etmeye başlar. Böylece havanın kırılma indisinde normal düzeyde olmayan mikro-ölçekli değişimler oluşmaya başlar. Bunlar da, elde edilen imgelerde homojen dağılmamış dalgalanmaların gözlenmesine neden olur [36].

Hem insanlar için, hem de bilgisayarlar için bir imge veya videoda bulunan sahnenin anlaşılabilir olması önemlidir. Sıcaklığın kendini iyice hissettirdiği yerlerde özellikle çöllerde ve otoyollarda, türbülansın güçlü etkilerini, görünür bir sıcak hava dalgası olarak imgelerde görmek mümkündür. Güçlü türbülans etkileri, bozulmuş imgede bütün sahnenin veya sahne içerisinde bazı bölgelerin anlaşılabilir olmasını zorlaştırır.

Elde edilen imgelerin görsel kalitesini arttırabilmek için öncelikli iş bozulmayı kestirebilmektir. Burada oluşan bozulmaları kestirebilmek için,

- 1) Matematiksel modelleme
- 2) Gözlem
- 3) Deney yöntemleri kullanılabilir [40].

Gözlem: Bozulma fonksiyonu hakkında herhangi bir bilginin olmadığı bir imge var olsun. Bu imgenin doğrusal ve uzamsal değişmeyen bir bozulma fonksiyonuna sahip

olduğu varsayılınsın. Bu bozulma fonksiyonunu kestirmenin yolu imgenin kendisinden bilgiyi almaktır. İmgede yüksek kontrastlı bir alt imge bölgesi aranıp, bulunan bu bölgeye imge iyileştirme uygulanarak orjinal sahnenin kestirimi sağlanır. Gözlenen alt imge bölgesi $I_g(x, y)$ ve işlenmiş alt imge bölgesi $I_f(x, y)$ ile tanımlansın. O zaman bozulma fonksiyonu:

$$\mathcal{H}_s(u, v) = \frac{\mathcal{G}_s(u, v)}{\mathcal{F}_s(u, v)}. \quad (2.1)$$

olur.

Deney : Eğer bozulmuş imgeleri toplamak için kullanılan ekipmana benzer bir ekipman mevcutsa bozulmanın bir kestirimi elde edilebilir. Ekipmandaki çeşitli sistem ayarlarıyla, verilen bozulmuş imgeye mümkün olan en benzer bozulmuş imgeler elde edilir. Daha sonra aynı sistem ayarları kullanılarak bir dürtünün (noktasal bir ışık kaynağı) fotoğraflanmasıyla bozulmanın dürtü tepkisi elde edilir. Buna göre bozulma fonksiyonu:

$$\mathcal{H}(u, v) = \frac{\mathcal{G}(u, v)}{\mathcal{A}}. \quad (2.2)$$

olur. Burada \mathcal{A} , dürtünün gücüdür.

Atmosferik türbülans kaynaklı bozulmalar genellikle zor modellenmektedir. İmgede homojen olmayan dalgalanmaların oluşması geometrik deformasyonlara neden olacağından, çok daha karmaşık hesaplamaları da beraberinde getirmektedir. Ayrıca modelleme yapılması için gereken havanın kırılma indisinin ve çevresel faktörlerin ölçümünde kullanılan cihazların maliyetinin yüksek olması bu alanda yapılan çalışmaları kısıtlamaktadır.

Yukarıda bahsedilen nedenlerden dolayı literatürde yapılan çalışmalarda genellikle bozulmaların kestirimi ya da modellenmesinden ziyade, imgelerin iyileştirilmesi hususunda çalışmalar mevcuttur.

Bu bölümde ilk önce, literatürde genel türbülans modellemesi için önerilen Kolmogorov türbülans teorisi, ardından çalışma boyunca kullanılacak imgelerin elde edilmesi işlemi anlatılacaktır. Daha sonra bu imgelerde bulunan bozulmaların görsel temsili PDM ve GKM yöntemleriyle sunulacaktır.

2.2 Kolmogorov Türbülans Teorisi

Atmosferdeki türbülans; havanın kırılma indisini hem uzamsal (spatial), hem de zamansal (temporal) olarak değiştiren sıcaklık ve konvektif hava hareketindeki rastgele değişimlerden kaynaklanır. Işık ışınları optik dalgalar şeklinde atmosferde yayıldıkça, kırılma indisindeki dalgalanmalar tarafından bozulur. Bu bozulmanın üstesinden gelmek için, türbülansın ve optik dalga yayılımı üzerine etkilerinin doğru bir şekilde modellenmesi gerekir.

Uzamsal ve zamansal tüm durumlar için kırılma indisini tam olarak tanımlamak imkansızdır. Kapalı form bir çözümde hesaba katılacak çok fazla rastgele hareket ve değişken mevcuttur. En yaygın olarak kabul edilen türbülanslı akış teorisi ilk olarak Kolmogorov [41] tarafından ortaya konulmuştur.

Obukhov [42] ve Corrsin [43], Kolmogorov'un türbülans modelini sıcaklık dalgalanmalarına uyarlamışlardır. Daha sonra, türbülanslı sıcaklık dalgalanmaları teorisi, kırılma indisi dalgalanmalarıyla doğrudan ilişkilendirilmiştir. Bu model, tüm türbülans teorilerinin temelini oluşturmaktadır [44].

Dünyanın güneş ışığına ve günlük döngüye göre farklı şekilde ısınması ve soğuması, hava sıcaklığında büyük ölçekli değişikliklere neden olur. Bu süreç sonuç olarak rüzgar yaratır. Hava hareket ettikçe laminer akıştan türbülanslı akışa geçer. Laminer akışta hız karakteristikleri eşittir veya en azından düzenli bir şekilde değişir. Türbülanslı akışta, farklı sıcaklıklardaki hava karışır, bu nedenle hız alanı artık tekdüze değildir ve türbülanslı girdaplar adı verilen rastgele dağıtılmış hava cepleri oluşur. Bu girdaplar değişen karakteristik boyutlara ve sıcaklıklara sahiptir. Havanın yoğunluğu ve dolayısıyla kırılma indisi sıcaklığa bağlı olduğundan, atmosferde rastgele bir kırılma indisi profili oluşur [45].

Türbülanslı akış, Navier-Stokes denklemleri tarafından temsil edilen doğrusal olmayan bir süreçtir. Tam gelişmiş türbülans için Navier-Stokes denklemlerini çözmeye zorluklar olduğundan, Kolmogorov istatistiksel bir teori geliştirmiştir. Türbülanslı akışta, büyük girdaplardaki kinetik enerjinin daha küçük girdaplara aktarılmasını önermiştir. En büyük girdapların ortalama boyutuna L_0 dış ölçek denir [45]. En küçük türbülanslı girdapların ortalama büyüklüğü l_0 , iç ölçek olarak adlandırılır. Çok küçük ölçeklerde, iç ölçekten daha küçük, sürtünmeden kaynaklanan enerji kaybı türbülansın kendini korumasını engeller. İç ölçek l_0 , zemin yakınlarında birkaç milimetre yüksekliğe ve zemin üzerinde birkaç santimetre yüksekliğe kadar çıkabilir.

Kolmogorov analizinde, ataletsel alt aralıktaki girdapların, alanın küçük bölgeleri içinde istatistiksel olarak homojen ve izotropik olduğunu, yani hız ve kırılma indisi gibi

özelliklerin sabit artışlara sahip olduğu varsaymıştır. Türbülanslı girdapların ortalama hızının (v), girdapların ölçek büyüklüğü (r) ile ilişkili olması gerektiğini göstermek için boyut analizi kullanmıştır [41]:

$$v \sim r^{1/3}, \quad (2.3)$$

Hızın yapı fonksiyonu $D_v(r)$, hızın karesi olduğundan; yapı fonksiyonu formu şu şekilde olmaktadır:

$$D_v(r) = C_v^2 r^{2/3}, \quad (2.4)$$

C_v^2 , hız yapı parametresidir. Çok küçük ölçeklerde oluşan laminar akış için fiziksel değişkenler biraz farklıdır, bu nedenle hız yapı fonksiyonu aşağıdaki gibi olur:

$$D_v(r) = C_v^2 l_0^{-4/3} r^2. \quad (2.5)$$

En büyük türbülans ölçekleri için akış oldukça anizotropiktir. Eğer hız alanı homojen ve izotropik olsaydı, yapı fonksiyonu asimptotik olarak hız varyansının iki katına yaklaşırdı.

Bu hız denklemleri, türbülanslı girdapların potansiyel sıcaklığına (Θ) da uyarlanabilir (potansiyel sıcaklık, normal sıcaklık \mathcal{T} ile doğrusal olarak ilişkilidir). Sonuç olarak $\Theta \sim r^{1/3}$, tür ve böylece potansiyel sıcaklık yapı fonksiyonu $D_\Theta(r)$, hız yapı fonksiyonu ile benzer değişkenlere sahiptir:

$$D_\Theta(r) = \begin{cases} C_\Theta^2 l_0^{-4/3} r^2, & 0 \leq r \ll l_0 \\ C_\Theta^2 r^{2/3}, & l_0 \ll r \ll \mathcal{L}_0 \end{cases} \quad (2.6)$$

Burada C_Θ^2 , Θ 'nın yapı parametresidir.

Kırılma indisindeki değişim, potansiyel sıcaklıktaki değişim ile doğru orantılı olduğundan; kırılma indisi yapı fonksiyonu $D_n(r)$, $D_\Theta(r)$ ile aynı güç yasasını izler. Böylece:

$$D_n(r) = \begin{cases} C_n^2 l_0^{-4/3} r^2, & 0 \leq r \ll l_0 \\ C_n^2 r^{2/3}, & l_0 \ll r \ll \mathcal{L}_0 \end{cases} \quad (2.7)$$

olur. Burada C_n^2 , kırılma indisi yapı parametresidir. Birimi $m^{-2/3}$ 'dür. Sıcaklık yapı sabiti ile ilgilidir. Matematiksel olarak şu şekilde ifade edilir:

$$C_n^2 = [77.6 \cdot 10^{-6} (1 + 7.52 \cdot 10^{-3} \Lambda^{-2}) \frac{\mathcal{P}}{\mathcal{T}^2}]^2 C_\Theta^2, \quad (2.8)$$

Burada Λ , mikrometre cinsinden dalga boyunu; \mathcal{P} , milibar cinsinden basıncı; \mathcal{T} ise Kelvin cinsinden sıcaklığı temsil etmektedir. C_n^2 'nin değişim aralığı 10^{-17} ila $10^{-13} m^{-2/3}$ dür ve yüksek irtifada küçük değerli, zemine yakın yerlerde büyük değerlidir.

Kırılma indisi yapı parametresi atmosferik türbülansın düzeyini belirtir. Günün hangi zaman diliminde olunduğuna, yerden yüksekliğine ve dünya üzerindeki lokasyonuna bağlı olarak değiştiğinden dolayı yerel türbülans düzeyini temsil eder.

Optik dalga yayılımında basınç ve nem gibi değişkenlerden daha çok sıcaklık değerinde oluşan değişimler kırılma indisi değerinde değişimlere neden olur. Bu nedenle basınç ve nem değerlerindeki değişimler genellikle ihmal edilir. Kırılma indisi değişimlerinin spektral bir tanıma ihtiyacı vardır. Böylece Kolmogorov kırılma indisi güç spektrum yoğunluğu aşağıdaki şekilde hesaplanır:

$$\Phi_n(\kappa) = 0.033 C_n^2 \kappa^{-11/3}, \quad \frac{1}{L_0} \ll \kappa \ll \frac{1}{l_0} \quad (2.9)$$

Burada κ açısal dalga sayısını temsil eder ve $2\pi/\Lambda$ değerine eşittir. Güç spektrum yoğunluğu tanımı sadece alt aralıktaki girdapların, alanın küçük bölgeleri içinde homojen ve izotropik olduğu durumlarda geçerlidir.

Atmosferik türbülans bozulmasının matematiksel bozulma modelini oluşturabilmek için kırılma indisi dalgalanmalarının genlik değerine (C_n^2) ihtiyaç vardır. Bu parametrenin ölçümü için gereken donanım temin edilemediği için bu çalışmada bu teoriden yararlanılmamıştır.

2.3 İmgelerin Elde Edilmesi

Atmosferik türbülansın etkileri genelde astronomik imgelerde ve uzak mesafeli görüntülemelerde daha baskındır. Bu da elde edilen imgelerde geometrik deformasyonlara ve bulanıklık etkilerinin gözlenmesine neden olmaktadır. Literatürde bu etkileri minimize etmeyi amaçlayan çalışmalar olsa da bu alanda çalışmaya açık, yaygın ve kullanışlı bir veri setini 2017 yılında Gilles [3] yayınladı ve kullanıma sundu. Elde ettiği imgelerde bulunan atmosferik türbülans etkileri, ısı kaynakları kullanılarak

yapay yollardan sağlanmıştır.

Veri seti konusunda paylaşımın sınırlı olmasından dolayı bu çalışmada öncelikli hedef, kendi veri setlerimizi elde edip bu imgeler üzerinde çalışmak olmuştur. Bütün veri setleri 2017 Ekim ve 2018 Temmuz aylarında Sky-Watcher marka BK MAK 127 AZ GT model teleskop ve Nikon marka D5100 model bir kamerayla elde edilmiştir. Görüntüleme sistemimizde bulunan teleskop ve kamera bileşenleri özellikleriyle beraber şu şekildedir;

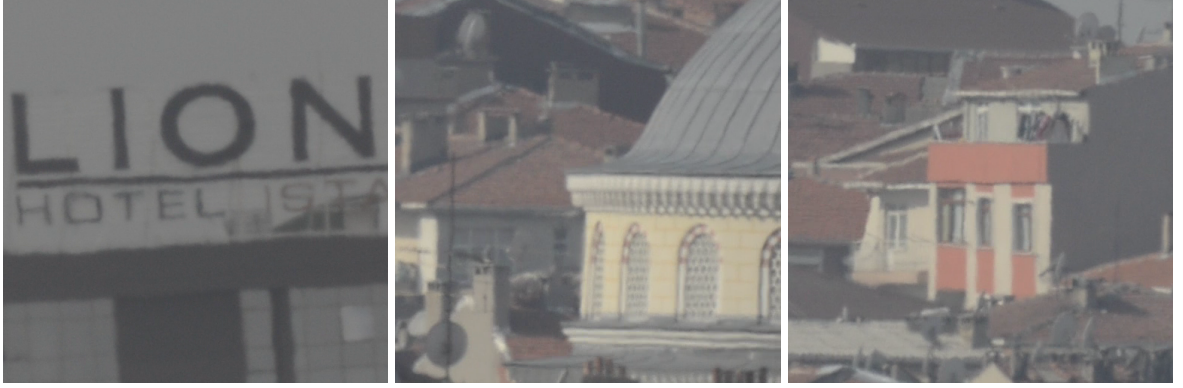
- **Teleskop:**Skywatcher marka 5 inç (127 mm) çaplı Maksutov-Cassegrain optik tasarıma sahip, odak uzaklığı 1500 mm, optik çapı:127 mm, F/oranı: F/11.8, maksimum büyütme oranı:254x ve çözünürlüğü 0.95 inç'tir.

- **Kamera:**Nikon marka D5100 model, etkin piksel sayısı 16.2 milyon imge boyutu 4928x3264, depolama dosya biçimi RAW veya JPEG'dir. Kamera 1080p çözünürlüğe ve saniyede 24 çerçeve oranına ayarlanmıştır. Elde edilen imgeler başlangıçta video olarak çekildiği için imgeler 4928x3264 boyutunda değil, 1920x1080 boyutundadır.

Elde edilen bütün imge setleri için Yıldız Teknik Üniversitesi Davutpaşa Kampüsü içerisinde farklı noktalardan ve kamera yerden yaklaşık 1.20 metre yükseklikteyken çekimler yapılmıştır.

Veri seti toplama işlemi bittikten sonra, kaydedilen videolar Windows işletim sistemi tabanlı bir bilgisayara yüklendikten sonra "ffmpeg" adlı yazılım programıyla (videoları ve imgeleri hemen her işletim sistemi üzerinden istenilen şekle çevirebilen kullanışlı bir program) videolar, PNG (portable network graphics- kayıpsız sıkıştırılmış imge saklama biçimi) uzantılı imge dizilerine çevrilmiştir. Orijinal imgenin ilgilenilen alanları (ROI-region of interest), bilgisayar terminalinde "convert" komutuyla istenilen şekle çevrilmiş ve PNG formatında imge seti olarak kaydedilmiştir.

Bu veri setinde 1-4. imge setleri Ekim 2017, 5-20. imge setleri Temmuz 2018 tarihleri arasında toplanmıştır. Bu veri tabanında toplam 20 adet veri seti bulunmaktadır. Elde edilen bütün imgelerde doğal ortamdan (hava sıcaklığı, rüzgarın şiddeti vb. gibi değişkenler manipüle edilmeden) elde edilmiştir. Bütün duruk imge setlerinden temsili imgeler, Şekil 2.1 ve Şekil 2.7 arasında verilmiştir.

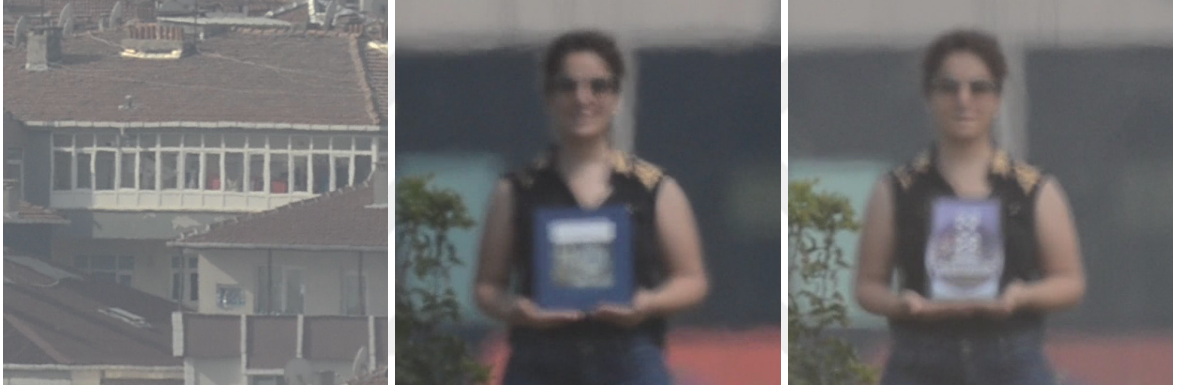


(a)

(b)

(c)

Şekil 2.1 Veri setlerinden temsili imgeler ((a) İmge Seti 1, (b) İmge Seti 2, (c) İmge Seti 3)



(a)

(b)

(c)

Şekil 2.2 Veri setlerinden temsili imgeler ((a) İmge Seti 4, (b) İmge Seti 5, (c) İmge Seti 6)



(a)

(b)

(c)

Şekil 2.3 Veri setlerinden temsili imgeler ((a) İmge Seti 7, (b) İmge Seti 8, (c) İmge Seti 9)



(a)

(b)

(c)

Şekil 2.4 Veri setlerinden temsili imgeler ((a) İmge Seti 10, (b) İmge Seti 11, (c) İmge Seti 12)



(a)

(b)

(c)

Şekil 2.5 Veri setlerinden temsili imgeler ((a) İmge Seti 13, (b) İmge Seti 14, (c) İmge Seti 15)



(a)

(b)

(c)

Şekil 2.6 Veri setlerinden temsili imgeler ((a) İmge Seti 16, (b) İmge Seti 17, (c) İmge Seti 18)



(a)



(b)

Şekil 2.7 Veri setlerinden temsili imgeler ((a) İmge Seti 19, (b) İmge Seti 20)



İmge setlerinin detayları şu şekildedir.

İmge Seti 1: 624x624 boyutlarında, 200 adet, Ekim 2017 tarihinde ve yaklaşık 3.5 km uzaklıktan çekilmiştir.

İmge Seti 2: 512x512 boyutlarında, 400 adet, Ekim 2017 tarihinde ve 500 m uzaklıktan çekilmiştir.

İmge Seti 3: 512x512 boyutlarında, 400 adet, Ekim 2017 tarihinde ve 600 m uzaklıktan çekilmiştir.

İmge Seti 4: 512x512 boyutlarında, 250 adet, Ekim 2017 tarihinde ve 500 m uzaklıktan çekilmiştir.

İmge Seti 5: 352x352 boyutlarında, 400 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 6: 352x352 boyutlarında, 600 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 7: 352x352 boyutlarında, 390 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 8: 400x400 boyutlarında, 485 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 9: 416x416 boyutlarında, 305 adet, Temmuz 2018 tarihinde ve 400 m uzaklıktan çekilmiştir.

İmge Seti 10: 528x528 boyutlarında, 305 adet, Temmuz 2018 tarihinde ve 700 m uzaklıktan çekilmiştir.

İmge Seti 11: 624x624 boyutlarında, 300 adet, Temmuz 2018 tarihinde ve 1 km uzaklıktan çekilmiştir.

İmge Seti 12: 624x624 boyutlarında, 300 adet, Temmuz 2018 tarihinde ve 1 km uzaklıktan çekilmiştir.

İmge Seti 13: 512x512 boyutlarında, 300 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 14: 704x704 boyutlarında, 470 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 15: 704x704 boyutlarında, 250 adet, Temmuz 2018 tarihinde ve 350 m uzaklıktan çekilmiştir.

İmge Seti 16: 592x592 boyutlarında, 635 adet, Temmuz 2018 tarihinde ve 2.5 km uzaklıktan çekilmiştir.

İmge Seti 17: 512x512 boyutlarında, 650 adet, Temmuz 2018 tarihinde ve 2.5 km uzaklıktan çekilmiştir.

İmge Seti 18: 512x512 boyutlarında, 810 adet, Temmuz 2018 tarihinde ve 1.4 km uzaklıktan çekilmiştir.

İmge Seti 19: 512x512 boyutlarında, 465 adet, Temmuz 2018 tarihinde ve 700 m uzaklıktan çekilmiştir.

İmge Seti 20: 512x512 boyutlarında, 205 adet, Temmuz 2018 tarihinde ve 700 m uzaklıktan çekilmiştir.

2.4 İmgelerdeki Bozulmaların Piksel Doluluğu Modelleme Kullanılarak Analizi

Hareket algılama yöntemlerinde amaç, imge dizilerinde zamanla oluşan değişikliklerin imge dizisi boyunca sabit olan bölgeden ayrılarak oluşan değişikliklerin tespit edilmesidir. Bu işlem yapılırken öncelikli olarak sabit olan arka planın çıkarılması gerekmektedir. Arka plan modelleme olarak bilinen bu işlem temel arka plan modelleme, istatistiksel arka plan modelleme, bulanık mantık arka plan modelleme ve arka plan kestirimi sınıflarına ayrılır [46]. Hareket eden nesne ve çevresi ile ilgili bir ön bilgi yoksa temel arka plan modelleme yöntemleri hareketin tespit edilmesi için (bu çalışma için bu konu bozulma bölgelerinin tespiti ve bozulmanın şiddeti ile ilgilidir) en sık kullanılan yöntemlerdir.

Değişimin olmadığı (yada çok az olduğu) bir imge (yaklaşık arka plan) olarak (1) imge dizilerinden biri arka plan modeli olarak seçilebilir, (2) bütün imgeler kullanılarak oluşturulan (zamansal ortalama veya zamansal medyan kullanılarak) bir arka plan modeli olabilir (3) yada başlangıç koşullarıyla başlatılıp yeğinlik değişimleriyle güncellenen bir model oluşturulabilir. Arka plan modelleme aynı zamanda verilen bir imge dizisinde benzerlik veya farklılıkların ölçülmesinde kullanılabilen yararlı bir araçtır. İmge dizisinden seçilen sıradaki imge, yukarıda bahsi geçen arka plan modellerinden biriyle oluşturulan arka plandan çıkarılır ve verilen bir piksel için elde edilen mutlak fark belirlenen bir eşik değerinden (Threshold - Th) büyükse o piksel ön plana ait bir bileşen olarak kabul edilir:

$$Z_k(x) = |I_k(x) - I_b(x)|, \quad (2.10a)$$

$$z_k(x) = \begin{cases} 1, & \text{Eğer } Z_k(x) > Th \\ 0, & \text{diğer durumlarda} \end{cases} \quad (2.10b)$$

x ; pikselin konumu, $k = 1, 2, 3, \dots, K$ olarak sıradaki imgenin indisi, I_k ; $m \times n$ boyutunda k .ncı gözlenen imge, I_b ; arka plan modeli, z_k ise değişen bölgeler için '1', durağan bölgeler için '0' değerini alan karar etiketidir. Eşik değerini belirlemek zorlu bir görevdir. Bu nedenle genelde deneysel olarak seçilir.

Eğer belli bir arka plan modeli yoksa yaklaşık arka plan, imge dizilerinin ortalaması veya medyanı olarak da seçilebilir. Medyan yönteminde imge dizisindeki K adet imge kaydedilir ve kaydedilen bu imgelerin medyanı hesaplanır. Ortalama yöntemi de aynı şekilde aynı noktaya karşılık gelen bütün piksel değerlerinin zamansal ortalaması

alınır ve arka plan modelinin ilgili piksel değeri olarak kabul edilir. Daha etkili bir yöntemde de gözlenen imgedeki bir pikselin değeri arka plan modelindeki piksel değerinden büyükse arka plan pikselinin değeri '1' arttırılır. Tam tersi durumda da '1' azaltılır [47].

Bu yaklaşımlar zaman içinde imge dizisini tarayarak hesaplandığından, hızlı olmasına rağmen her defasında imgenin boyutları ve imge sayısının çarpımı kadar hafızaya ihtiyaç duyacaktır. Ardışık imgeler kullanılarak yapılacak küçük bir değişiklikte:

$$Z_{k,r}(\mathbf{x}) = |I_k(\mathbf{x}) - I_r(\mathbf{x})|, \quad (2.11a)$$

$$z_{k,r}(\mathbf{x}) = \begin{cases} 1, & \text{Eğer } Z_{k,r}(\mathbf{x}) > Th \\ 0, & \text{diğer durumlarda} \end{cases} \quad (2.11b)$$

I_r ; referans imgeyi temsil etmektedir ve $Z_{k,r}(\mathbf{x})$ 'in ardışık iki imgenin farkı olabilmesi için I_{k-1} alınır. Bu senaryoda yaklaşık arka plan bir önceki imgedir ve verilen piksel için piksel değerleri farkı eşik değerinden büyükse ön planın bir parçası olarak atanır. Eğer imge dizileri arasında büyük zamansal/uzamsal farklar yoksa temel arka plan modelleme çok iyi sonuç verecektir. Bu yöntemin en önemli avantajı gürültüyü bastırmasıdır. Yaklaşık arka plan modeli bir önceki imge kabul edildiği için, arka plandaki değişikliklere diğer yöntemlerden daha hızlı adapte olur. Başka bir değişiklik de [48]:

$$I_{b_k}(\mathbf{x}) = \alpha \cdot I_{k-1}(\mathbf{x}) + (1 - \alpha) \cdot I_{b_{k-1}}(\mathbf{x}) \quad I_{b_0} = [0], \quad (2.12a)$$

$$Z_k(\mathbf{x}) = |I_k(\mathbf{x}) - I_{b_k}(\mathbf{x})|, \quad (2.12b)$$

$$z_k(\mathbf{x}) = \begin{cases} 1, & \text{Eğer } Z_k(\mathbf{x}) > Th \\ 0, & \text{diğer durumlarda} \end{cases} \quad (2.12c)$$

I_{b_k} ; k .nci imge için yaklaşık arka planı, α öğrenme oranını (genelde 0.05 alınır) temsil etmektedir. Bu yöntemde imge dizisinden iki ardışık imge kullanıldığı için bütün imgeleri saklamaya gerek yoktur. Eğer α 1'e eşit olursa, $I_{b_k}(\mathbf{x}) = I_{k-1}(\mathbf{x})$ olur ve bu da ardışık imge farkı yöntemini verir.

2.4.1 Deneysel Sonuçlar

Atmosferik türbülansın zamansal/uzamsal değişen etkileri imgede yerel dalgalanmalar şeklinde gözleneceğinden imge dizisindeki her imgede bozulmaya uğrayan bölgeler ve bozulmanın şiddeti değişiklik gösterecektir. Bu nedenle hareket algılama yöntemleri bozulmaların bulunduğu bölgeleri görsel olarak ifade etmekte kullanılabilir.

Bu senaryoda görüntüleme sistemi ve görüntülenen sahne sabittir. Elde edilen duruk imge dizileri arasındaki zamansal fark, imgede gölge veya herhangi başka bir değişkenin bozulma bölgelerinin tespitini zora sokacak şekilde bir aydınlatma değişimi söz konusu değildir. İmge dizileri duruk olduğundan dolayı arka plan modellemesi güncellenmemiştir. Yapılan çalışmada öncelikle imge dizilerinden bir veri küpü oluşturulmuştur. Ardından verilen herhangi bir piksel için o noktada bulunan imge sayısı kadar değer arasından maksimum ve minimum değerler arasındaki fark alınmıştır ve kaydedilmiştir. Bu değerler aynı piksel noktasına kaydedilmiştir. Kaydedilen bu değerler arasındaki tepe değerinin 0.2 katı, eşik değeri olarak belirlenmiştir. Eşik değeri seçimi deneysel olarak belirlenmiştir. Bir sonraki aşamada her bir piksel için o pikselin içindeki bütün değerlerin sırasıyla bahsedilen pikselin medyan değeriyle farkı alınır. Bu değer belirlenen eşik değerinden büyükse başlangıçta 0 değeriyle tanımlanan bir değişkenin değeri 1 arttırılır. Bu şekilde o piksel için bu işlem imge sayısı kadar tekrarlanır ve son sayı kaydedilir. Burada yapılan işlem her bir imgenin bahsi geçen pikseli için eşik değerini kaç defa aştığına bağlı olarak bir bozulma şiddeti tanımlamaktır. İşlem bittikten sonra her bir pikselde bulunan değer imge sayısı ile karşılaştırılır ve lineer bir model oluşturularak bölgede bulunan bozulma şiddeti görsel olarak tanımlanır. Yöntem şu şekilde ifade edilir:

$$Z(x) = \max(I(x)) - \min(I(x)), \quad (2.13a)$$

$$Th = 0.2 * (\max(Z(x))) \quad (2.13b)$$

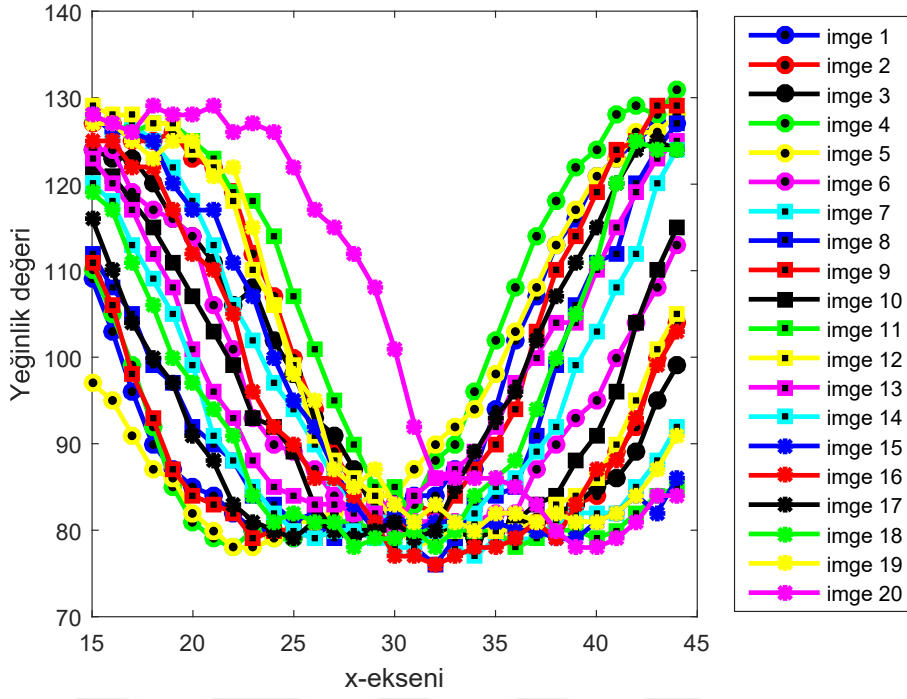
$$c(x) = \begin{cases} c_k(x) + 1, & \text{Eğer } (I_k(x) - I_{med}(x)) > Th \\ c_k(x), & \text{diğer durumlarda} \end{cases} \quad (2.13c)$$

$$z(x) = \begin{cases} \text{Siyah bölge,} & \text{Eğer } c(x) \leq \frac{K}{4} \\ \text{Mavi bölge,} & \text{Eğer } \frac{K}{4} < c(x) \leq \frac{K}{2} \\ \text{Turkuaz bölge,} & \text{Eğer } \frac{K}{2} < c(x) \leq \frac{3K}{4} \\ \text{Beyaz bölge,} & \text{diğer durumlarda} \end{cases} \quad (2.13d)$$

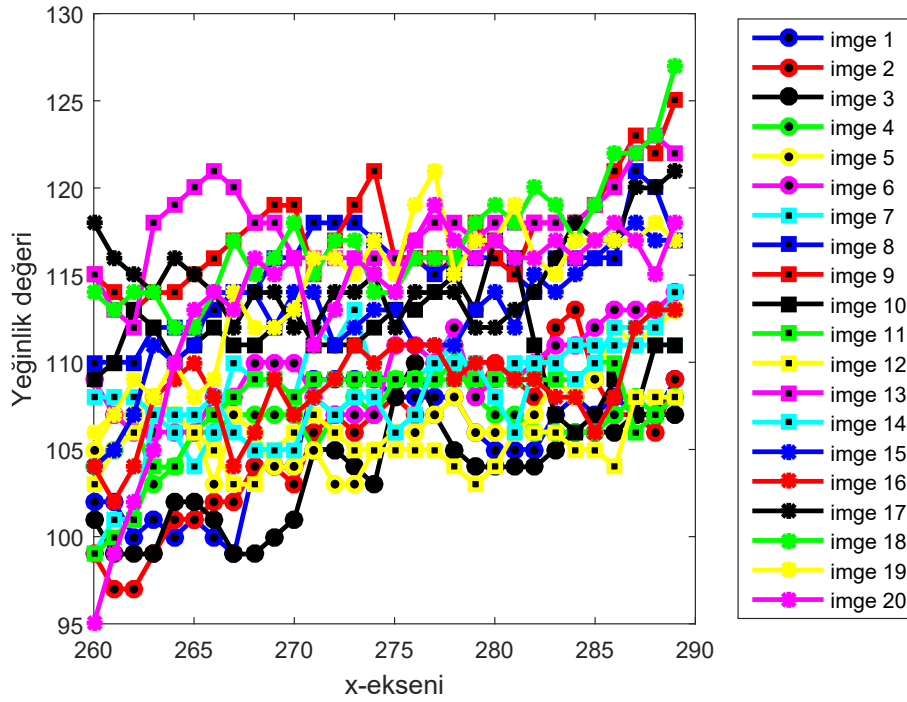
Burada c artan etiket, K ise toplam imge sayısını göstermektedir. İmge dizisinde bulunan bozulmanın şiddeti 4 sınıfa ayrılmıştır. Şekil 2.12'den Şekil 2.20'ye kadar olan figürler veri setinde bulunan belirli imge dizileri için bu yöntemle ortaya çıkan görsel temsil sonuçlarıdır. Şekil 2.8'de bozulma şiddetini farklı bölgelerde görmek adına İmge seti-1 için belirli bölgeler kırmızı ((15,200) ve (44,200) pikselleri arasında kalan bölge- toplam 30 piksel), yeşil ((260,315) ve (289,315) pikselleri arasında kalan bölge- toplam 30 piksel) ve mavi ((315,335) ve (324,335) pikselleri arasında kalan bölge- toplam 10 piksel) renklerle belirtilmiştir. Bu bölgelerde atmosferik türbülans bozulmasının piksellerde yol açtığı yeğlilik değerinde oluşan değişimler Şekil 2.9'da kırmızı bölge için, Şekil 2.10'da yeşil bölge için ve Şekil 2.11'de ise mavi bölge için gösterilmiştir.



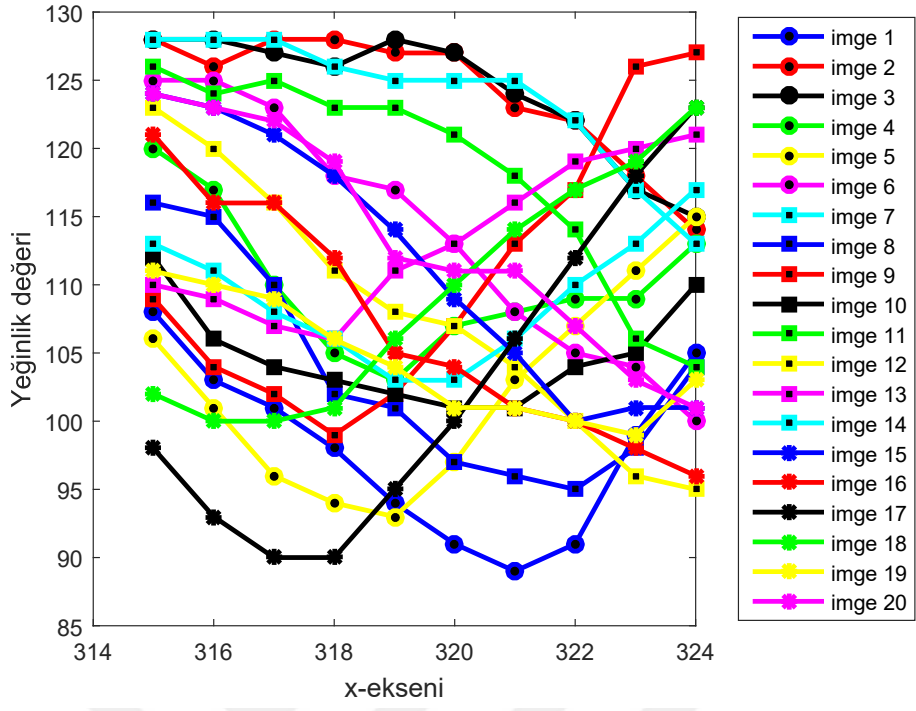
Şekil 2.8 Veri seti-1 için farklı alanlarda yeğlilik değişimlerinin belirlenmesi



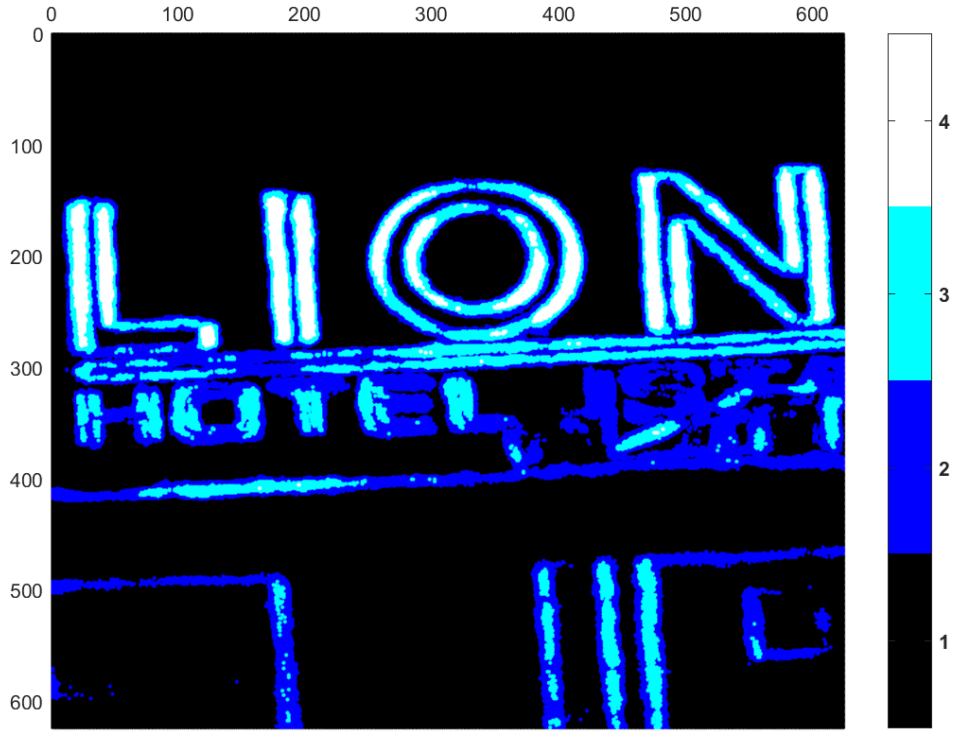
Şekil 2.9 Veri seti-1 Şekil 2.8’de görülen kırmızı çizgi için imge dizisi yeğinlik değişimi



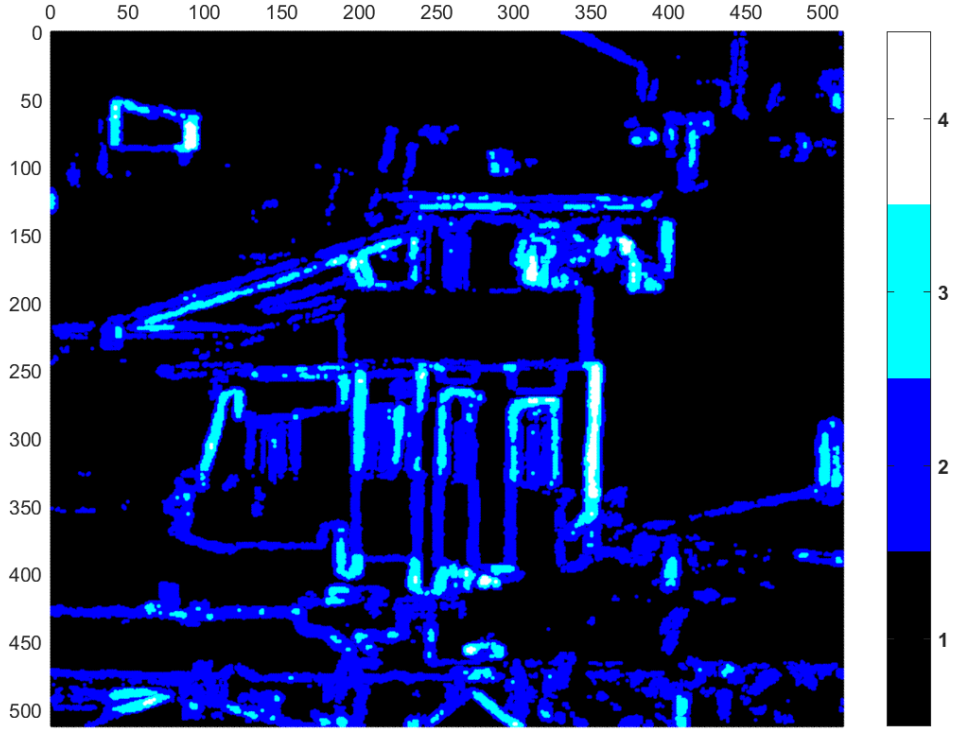
Şekil 2.10 Veri seti-1 Şekil 2.8’de görülen yeşil çizgi için imge dizisi yeğinlik değişimi



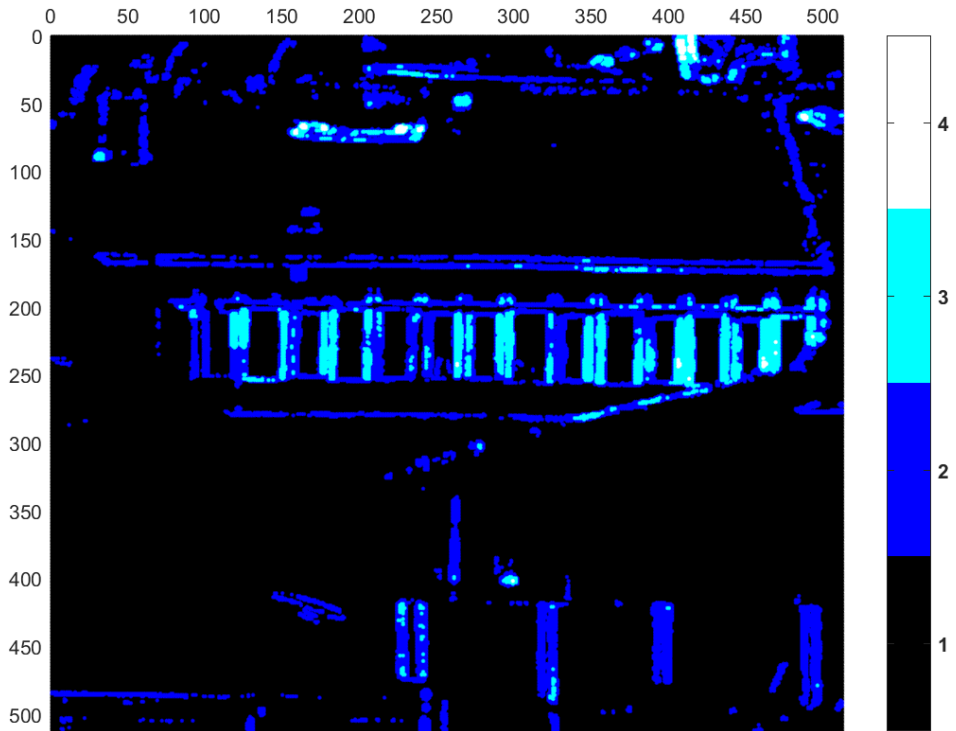
Şekil 2.11 Veri seti-1 Şekil 2.8’de görülen mavi çizgi için imge dizisi yeğİnlik değİşimi



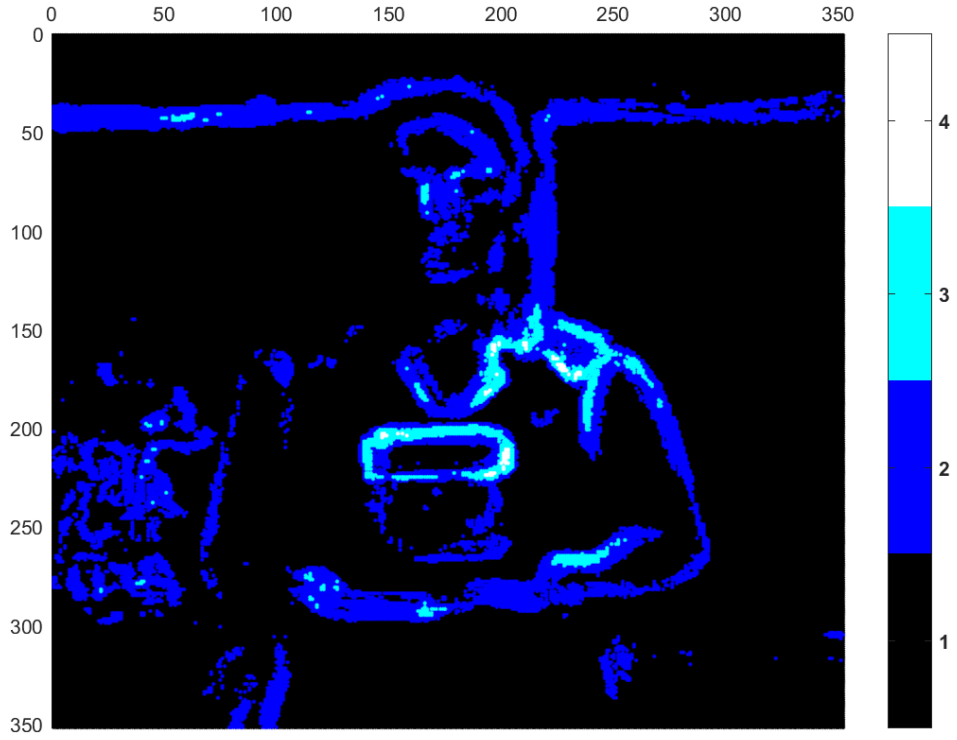
Şekil 2.12 PDM yöntemiyle Veri seti-1 için bozulmaların görsel temsili



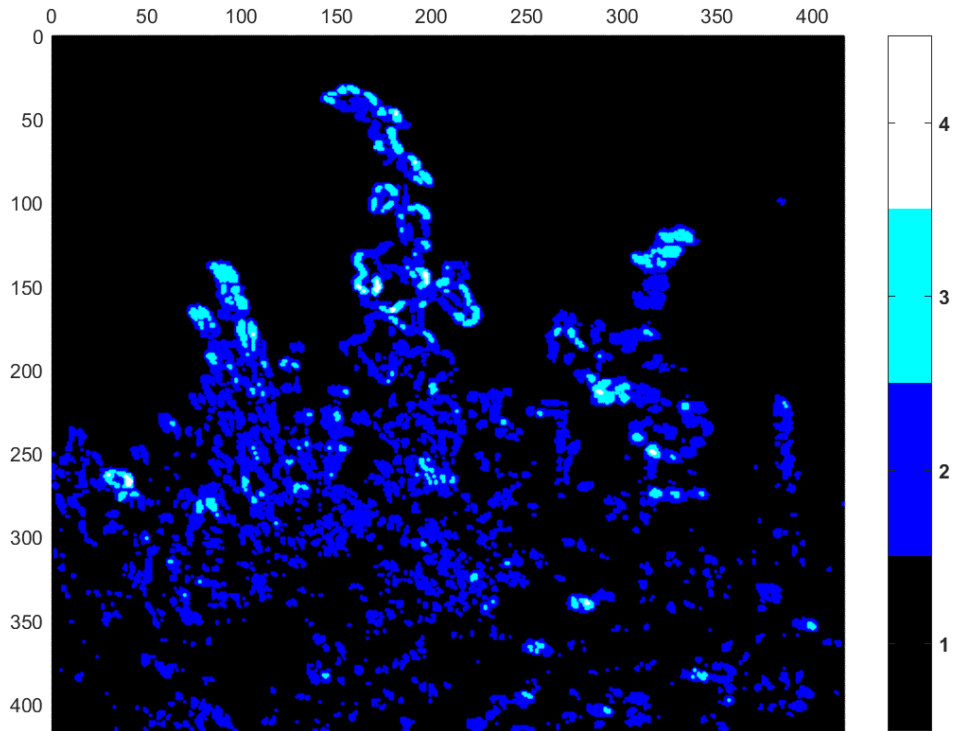
Şekil 2.13 PDM yöntemiyle Veri seti-3 için bozulmaların görsel temsili



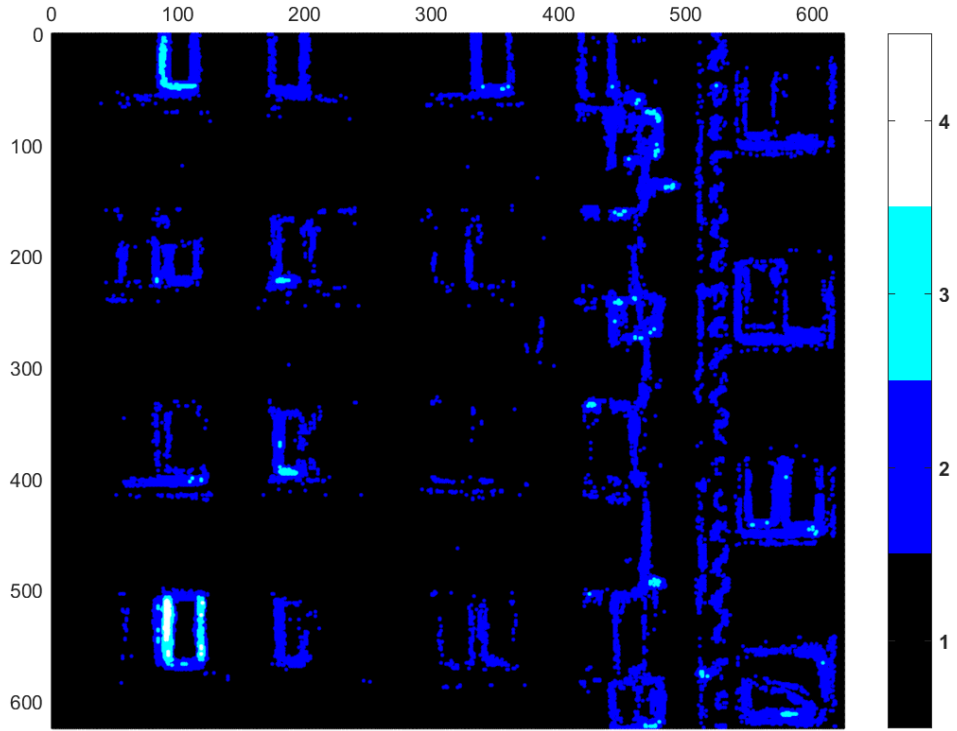
Şekil 2.14 PDM yöntemiyle Veri seti-4 için bozulmaların görsel temsili



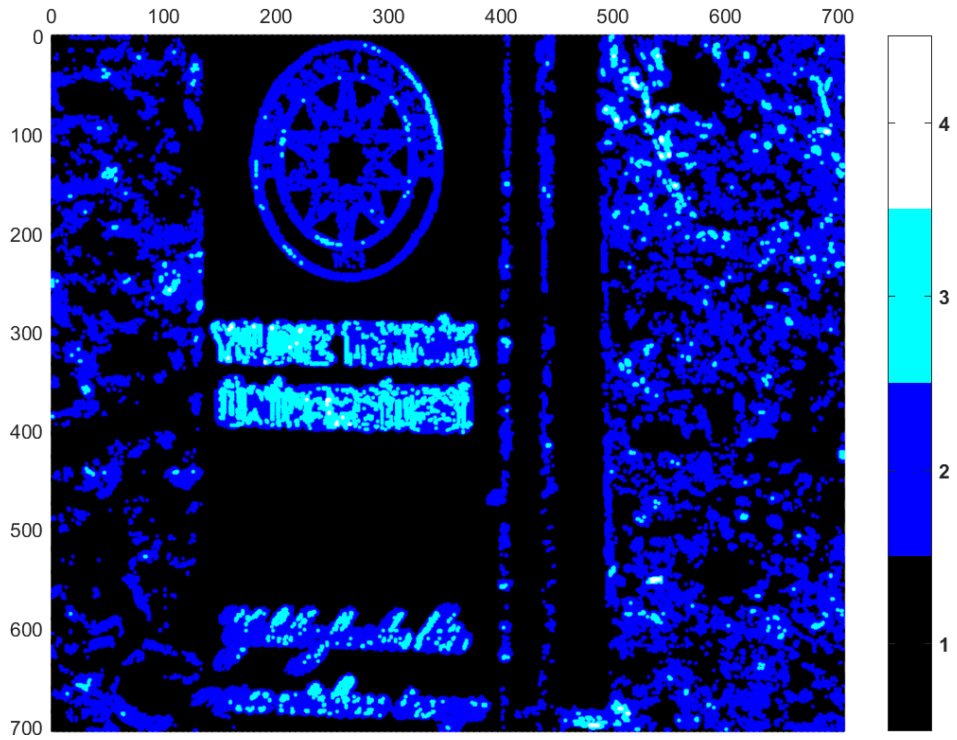
Şekil 2.15 PDM yöntemiyle Veri seti-5 için bozulmaların görsel temsili



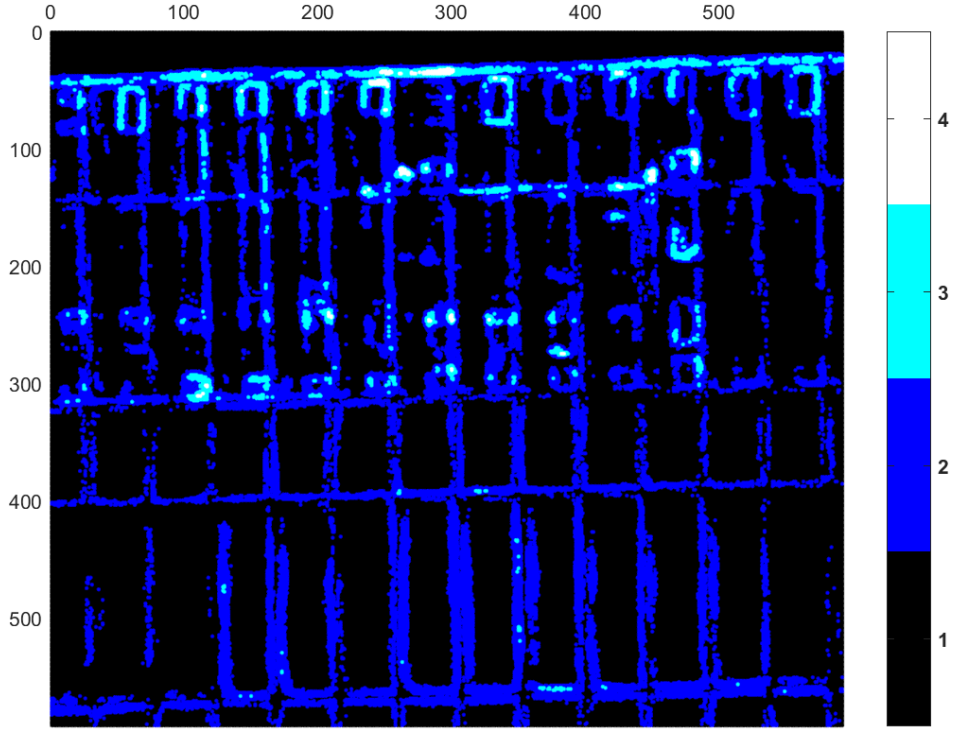
Şekil 2.16 PDM yöntemiyle Veri seti-9 için bozulmaların görsel temsili



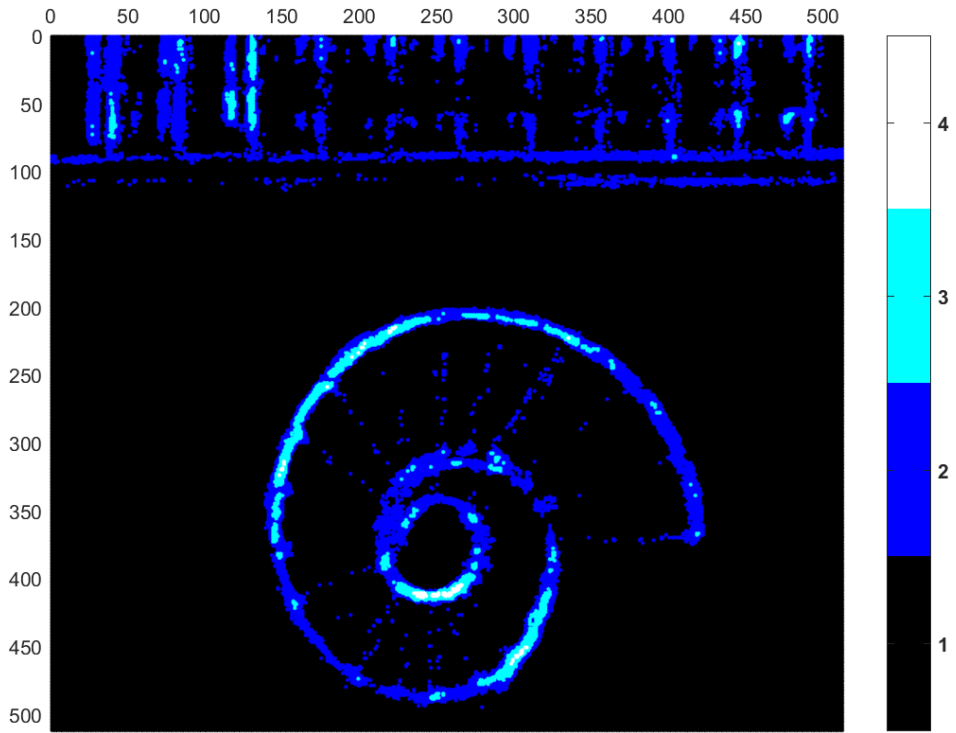
Şekil 2.17 PDM yöntemiyle Veri seti-12 için bozulmaların görsel temsili



Şekil 2.18 PDM yöntemiyle Veri seti-15 için bozulmaların görsel temsili



Şekil 2.19 PDM yöntemiyle Veri seti-16 için bozulmaların görsel temsili

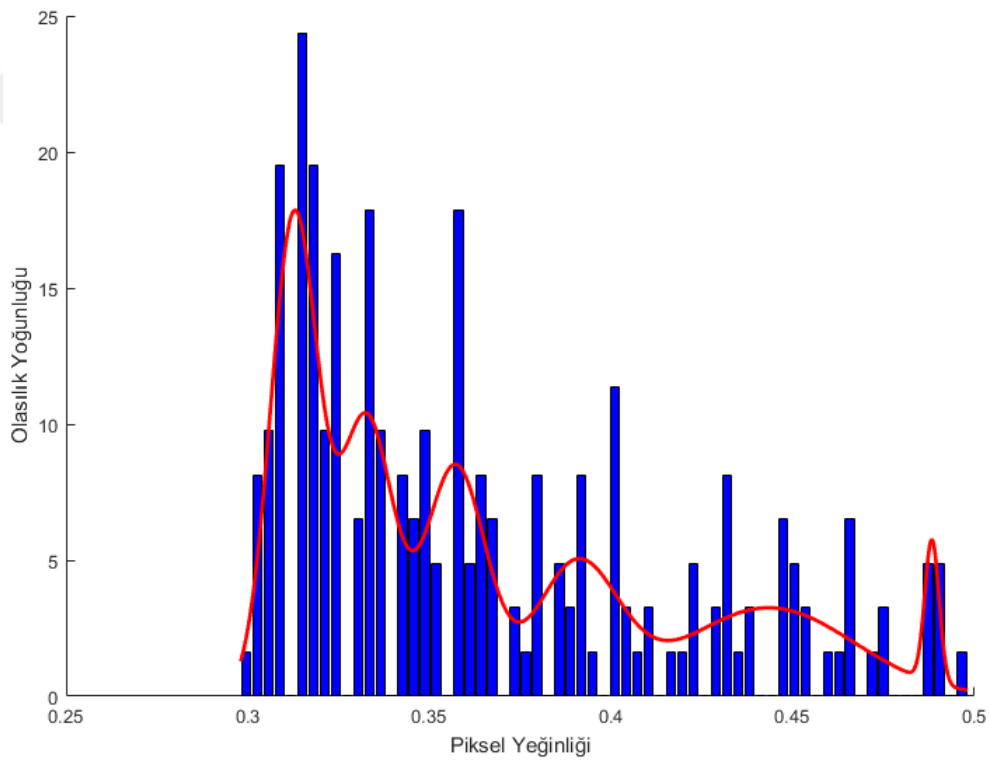


Şekil 2.20 PDM yöntemiyle Veri seti-17 için bozulmaların görsel temsili

2.5 İmgelerdeki Bozulmaların Gauss Karışım Modeli Kullanılarak Analizi

İmgelerdeki deformasyona uğramış bölgelerin tespitinde kullanabileceğimiz bir diğer model ise GKM'dir [49]. Mevcut hareket algılama yöntemleri içinde en popüler yöntemdir ve genellikle kümelenendirme veya sınıflandırma işlemlerinde kullanılır. Bu nedenle imgede bulunan bozulmaya uğramamış bölgeleri deforme olmuş bölgelerden bu yöntemle ayırabiliriz.

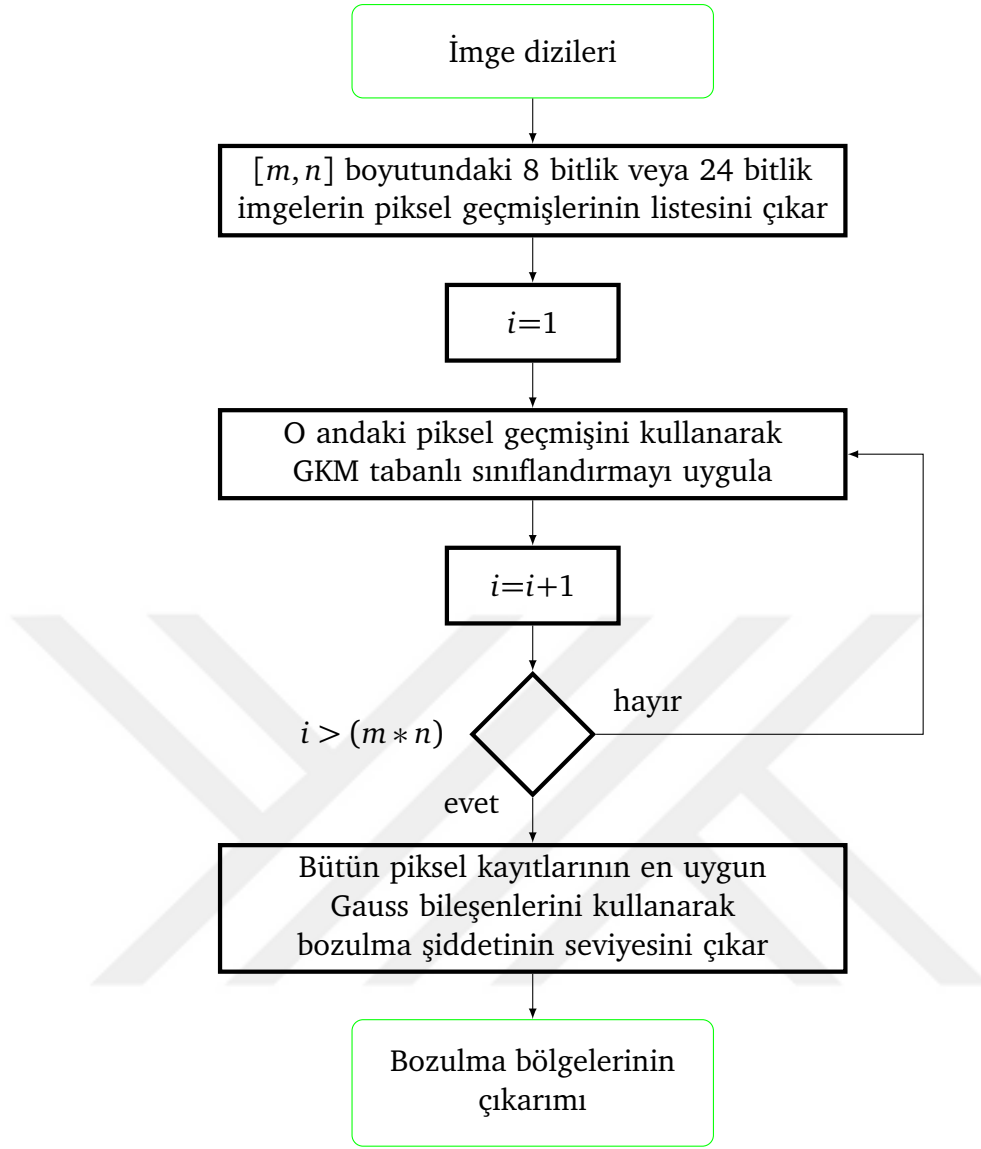
Şekil 2.21'de veri seti-1 için (200,200) noktasında imge dizisinin GKM gösterimi sunulmuştur.



Şekil 2.21 Veri Seti-1'de (200,200) noktası için GKM gösterimi

Şekil 2.22'de deneysel çalışmada kullanılan yöntemin blok diyagramı görölmektedir. Bir imgede piksel deđerlerini doğrudan karşılaştırmanın bir yöntemi de piksellerin istatistiklerini incelemektir. İmge dizisindeki herhangi bir anda her pikselin katkısının anlaşılabilmesi için, bu piksellere bir ađırlık tanımlanmalıdır.

Piksel geçmişı $X \in \mathbb{R}^{(N \times p)}$ (p boyutlu, N nokta) verilen bir veride; K adet grubun, kümenin veya sınıfın sayısını tahmin etmekle ilgilenilebilir. Gözlenen her piksel geçmişinin sınıf üyeliđi ($\hat{y}_n|X, n = 1 \dots N, \hat{y}_n \in \{1, \dots, K\}$) dır. Burada \hat{y}_n ; bunun kesin bir sınıf üyeliđi deđil, tahmini bir sınıf üyeliđi olduđunu belirtir. GKM bu senaryoda, verilen verinin karışım olasılık yođunluk fonksiyonuna uymasına yardım etmek için



Şekil 2.22 GKM yöntemiyle bozulma bölgelerinin gösterimi

iyi bir araçtır. Ayrıca bu yöntem, tahmin ve optimizasyon gibi istatistiksel işlemlerin de ilerletilmesine izin verir. Her biri farklı bir sınıfa ait K 'nın bir karışımından gelen p -boyutlu gözlemleri ($x_n | n = 1 \dots N$) varsayarsak, karışım yoğunluğunu şu şekilde ifade edebiliriz:

$$f(\mathbf{x}; w, \theta) = \sum_{k=1}^K w_k \eta(\mathbf{x}; \theta_k) \quad (2.14)$$

Burada w_k karışım oranlarıdır ve $w_k > 0$, $\sum_{k=1}^K w_k = 1$, θ_k ise k . bileşenin bilinmeyen parametrelerinin vektörü ve η da, bir gözleme ait k . bileşenin olasılığını temsil etmektedir. $\theta_k = (\mu_k, \Sigma_k)$ parametreleriyle çok değişkenli η fonksiyonu;

$$\eta(\mathbf{x}; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k)\right) \quad (2.15)$$

Sınıflandırma problemine bu olasılıksal açıdan yaklaşılması, tüm sorunu bir karışım yoğunluğunun parametre tahminine indirger. Denklem 2.15'te verilen Gauss karışım yoğunluğunun bilinmeyen parametreleri; karışım oranları w_k , ortalama vektörler μ_k ve kovaryans matrisler Σ_k 'dir. Bu parametreleri kestirmek için, parametre tahminine olan güvenin en üst düzeye çıkarılması gerekmektedir. Veriler karışım oranıyla ağırlıklandırılmış tek bir bileşen için parametrelerin olasılık fonksiyonu olarak ifade edileceğinden:

$$f(x_n; w_k, \theta_k) = w_k \eta(x_n; \theta_k), \quad (2.16a)$$

$$L(\theta_k | X) = f(x_1, x_2, \dots, x_N; w_k, \theta_k) L(\theta_k | X) = f(x_1; w_k, \theta_k) f(x_2; w_k, \theta_k) \dots f(x_N; w_k, \theta_k) \quad (2.16b)$$

$$L(\theta_k | X) = \prod_{n=1}^N w_k \eta(x_n; \mu_k, \Sigma_k) \quad (2.16c)$$

Denklem 2.16'da verilen terimin logaritmasını almak, tek bir bileşenle ilgili en çok benzerlik olasılığını verir ve toplam K adet bileşen olduğu için GKM'nin en üst seviye

benzerliği:

$$\begin{aligned} \log L(\theta_k|X) &= \sum_{n=1}^N \log \{w_k \eta(x_n; \theta_k)\} \\ \log L(\theta|X) &= \sum_{k=1}^K \sum_{n=1}^N \log \{w_k \eta(x_n; \mu_k, \Sigma_k)\} \end{aligned} \quad (2.17)$$

Herhangi bir karışım dağılımı için $\log L(\theta|X) = 0$ kapalı form çözümü yoktur. Bu nedenle en çok benzerlik sayısal olarak en yüksek olmak zorundadır. Bu sayısal optimizasyon için de genelde beklenti-embüyütme yöntemi (Expectation Maximization) kullanılır.

Sayısal Optimizasyon İçin Beklenti-Embüyütme Algoritması

Beklenti-embüyütme algoritması, aşağıdaki parametreler için bazı başlangıç değerleri verildiğinde, iki alternatif adımdan oluşan yinelemeli bir işlemdir. Bunlar:

$$\hat{w}_k = \frac{1}{N} \sum_{n=1}^N Cl_k(\hat{y}_n), \quad \hat{\mu}_k = \frac{1}{\hat{w}_k N} \sum_{n=1}^N x_n Cl_k(\hat{y}_n) \quad (2.18a)$$

$$\hat{\Sigma}_k = \frac{1}{\hat{w}_k N} \sum_{n=1}^N [(x_n - \hat{\mu}_k)^T (x_n - \hat{\mu}_k)] Cl_k(\hat{y}_n) \quad (2.18b)$$

$$Cl_k(\hat{y}_n) = \begin{cases} 1 & \hat{y}_n = k \\ 0 & \hat{y}_n \neq k \end{cases} \quad (2.18c)$$

Beklenti-embüyütme algoritması, Denklem 2.18'deki parametreler için bazı başlangıç değerleri verildiğinde, iki alternatif adımdan oluşan yinelemeli bir prosedürdür. $Cl_k(\hat{y}_n)$ değerleri, bir başlatma kalıbı (örneğin K-en yakın komşuluk kümelemesi) ile hesaplanır. Beklenti-embüyütme algoritmasının yinelemeli adımları aşağıdaki gibi özetlenebilir:

- 1) Başlangıç parametreleri $\hat{w}_k^{(t)}$, $\hat{\mu}_k^{(t)}$, $\hat{\Sigma}_k^{(t)}|_{t=0}$ ile başla
- 2) $t = 0 \dots T$ için veya bir durdurma kriteriyle karşılaşılan kadar beklenti adımını ve embüyütme adımını yinelemeli olarak uygula.

$(t \geq T)$ ve $(|(\log L^{(t+1)}(\hat{\theta}|X) - \log L^{(t)}(\hat{\theta}|X)) / (\log L^{(iter)}(\hat{\theta}|X))| > Ltol)$

2.1. Beklenti Adımı– k . bileşenin n . gözlemine ait sonsal olasılık $\hat{\tau}_{n,k}$, önceki parametre tahminleri kullanılarak tahmin edilir:

$$\hat{\tau}_{n,k} = \frac{\hat{w}_k^{(t)} \eta(x_n; \hat{\mu}_k^{(t)}, \hat{\Sigma}_k^{(t)})}{\sum_{k=1}^K \hat{w}_k^{(t)} \eta(x_n; \hat{\mu}_k^{(t)}, \hat{\Sigma}_k^{(t)})} \quad (2.19)$$

2.2. Enbüyütme Adımı– w_k , μ_k ve Σ_k parametre tahminleri, verilen tahmini sonsal olasılıklar ile güncellenir:

$$\hat{w}_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^N \hat{\tau}_{n,k} \quad (2.20a)$$

$$\hat{\mu}_k^{(t+1)} = \frac{1}{N \hat{w}_k^{(t+1)}} \sum_{n=1}^N x_n \hat{\tau}_{n,k} \quad (2.20b)$$

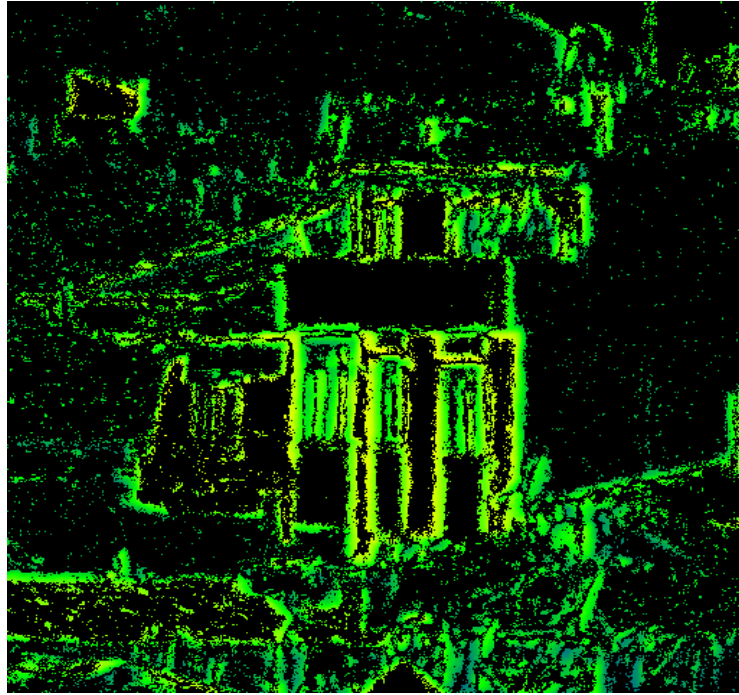
$$\hat{\Sigma}_k^{(t+1)} = \frac{1}{N \hat{w}_k^{(t+1)}} \sum_{n=1}^N [(x_n - \hat{\mu}_k^{(t+1)})^T (x_n - \hat{\mu}_k^{(t+1)})] \hat{\tau}_{n,k} \quad (2.20c)$$

2.5.1 Deneysel Sonuçlar

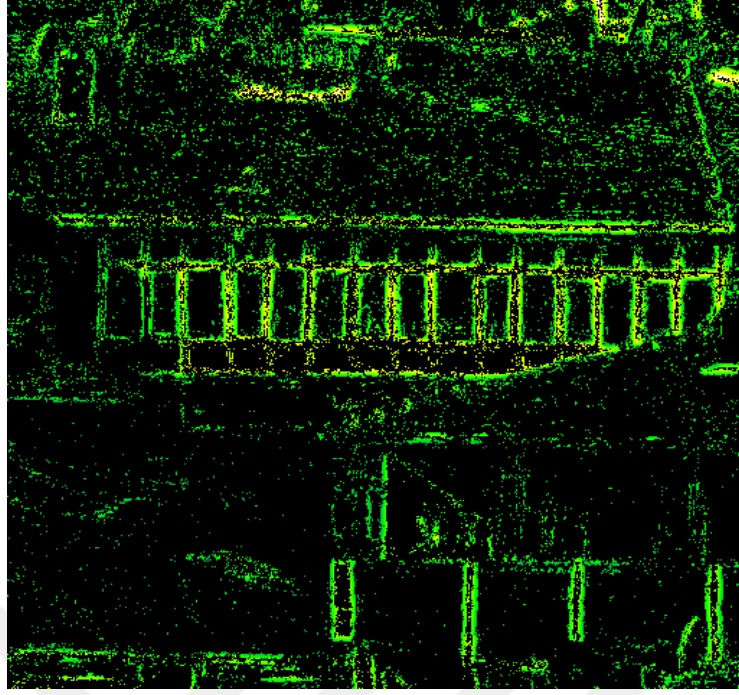
Bu bölümde tıpkı PDM bölümünde olduğu gibi aynı imge setleri kullanılmış ve bu setlere GKM yöntemi uygulanarak bozulmaların hangi bölgelerde bulunduğu görülmektedir. Şekil 2.23 ile Şekil 2.31 arasındaki figürler belirtilen veri setleri üzerinde GKM yönteminin sonuçlarını göstermektedir. Genelde GKM için seçilen bileşen sayısı manuel olarak 3 ila 5 arasında değişmektedir. Bileşen sayısı PDM’de seçilen değer ile aynı tutularak 4 alındı. Parametre değerleri en iyi performansı gösterecek şekilde ayarlandı. İki yöntemde de bozulma bölgelerinin tespiti ve bozulma şiddeti için oluşturulan haritalarda yüksek oranda benzerlik bulunmaktadır.



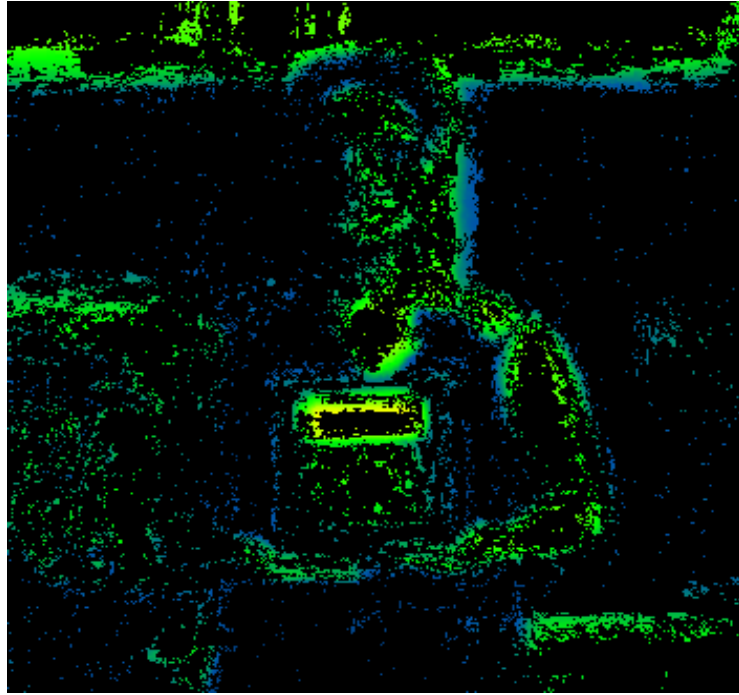
Şekil 2.23 GKM yöntemiyle Veri seti-1 için bozulmaların görsel temsili



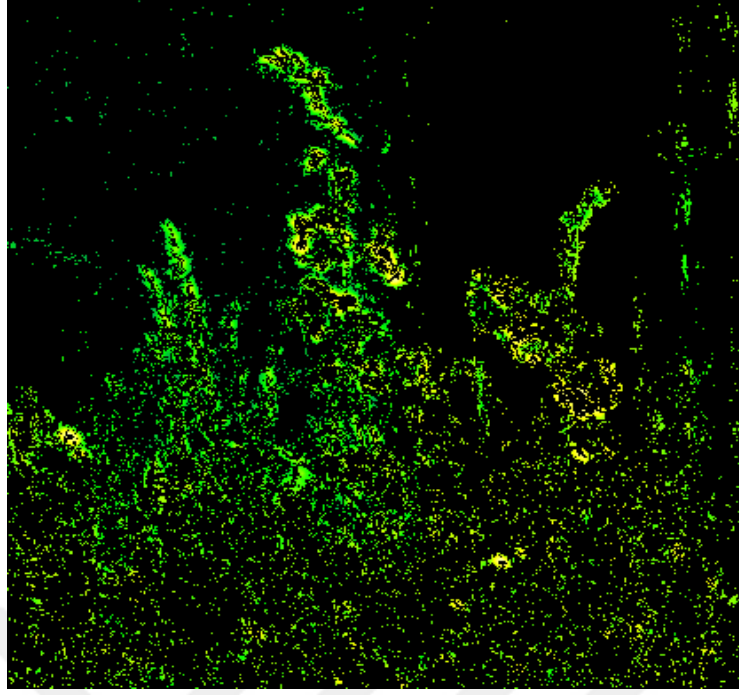
Şekil 2.24 GKM yöntemiyle Veri seti-3 için bozulmaların görsel temsili



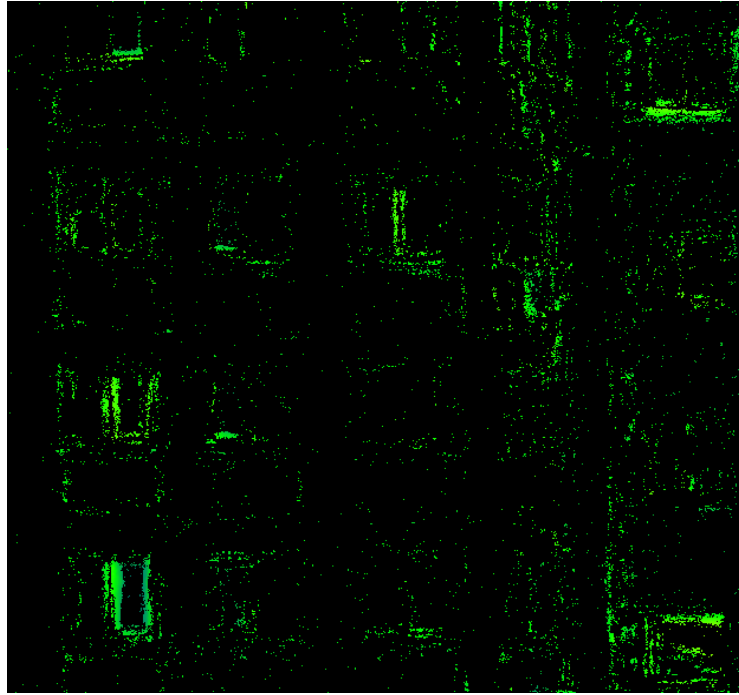
Şekil 2.25 GKM yöntemiyle Veri seti-4 için bozulmaların görsel temsili



Şekil 2.26 GKM yöntemiyle Veri seti-5 için bozulmaların görsel temsili



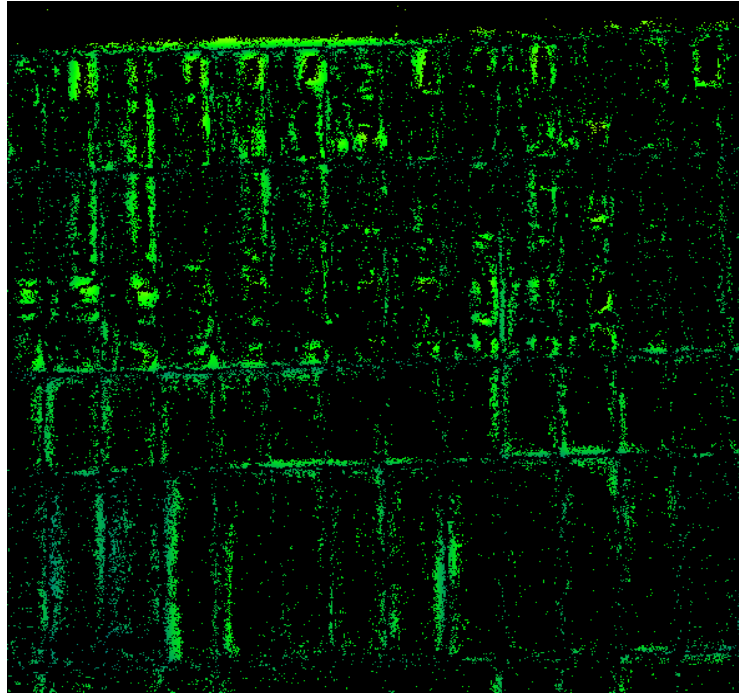
Şekil 2.27 GKM yöntemiyle Veri seti-6 için bozulmaların görsel temsili



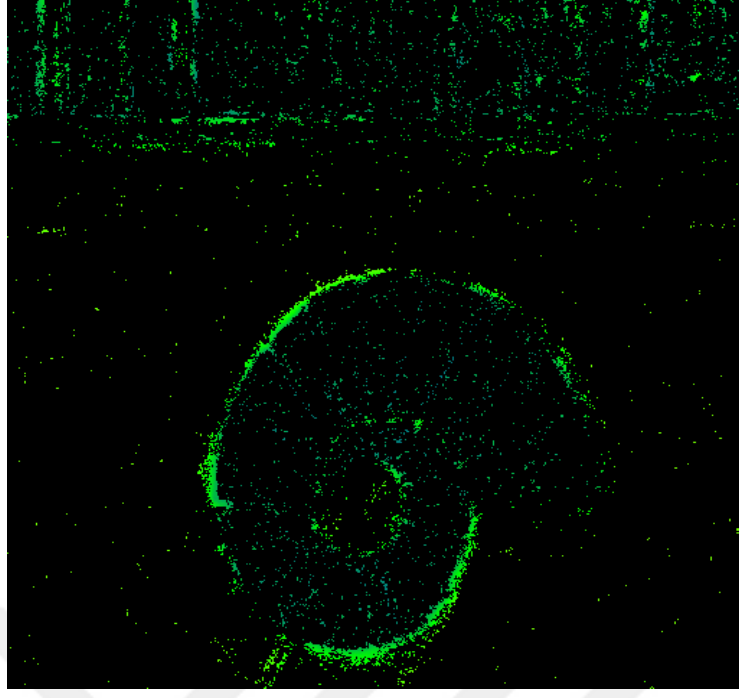
Şekil 2.28 GKM yöntemiyle Veri seti-12 için bozulmaların görsel temsili



Şekil 2.29 GKM yöntemiyle Veri seti-15 için bozulmaların görsel temsili



Şekil 2.30 GKM yöntemiyle Veri seti-16 için bozulmaların görsel temsili



Şekil 2.31 GKM yöntemiyle Veri seti-17 için bozulmaların görsel temsili

UZAK MESAFELİ GÖRÜNTÜLEMEDE İMGE ÇAKIŞTIRMA

İmge veya videolarda atmosferik türbülans görsel kaliteyi olumsuz yönde etkiler. Güçlü türbülans etkisi altındaki ortamda bu etkiler, elde edilen imgede yer yer geometrik deformasyon ve bulanıklık şeklinde görülebilir.

Atmosferik türbülans etkilerini azaltmak, çözümü zor bir problemdir. İmgede bulunan geometrik deformasyonlardan dolayı tek bir imge kullanımını orjinal sahnenin kestirimi için yeterli bilgiyi içermeyecektir. Bu nedenle bozulmalardan kurtulmak için etkili bir yöntem değildir.

İmge çakıştırma, geometrik deformasyon ve zamanla değişen bulanıklık bozulmalarının minimize edilmesinde imge dizilerine uygulanan iyileştirme algoritmalarının en önemli adımlarından biridir. Orjinal sahneye yakın bir imge kestirimine sahip olmak için, etkin imge çakıştırma yönteminin kullanılması gerekir. Bu işlem, orta seviye imge işleme görevlerinden biridir ve temel amacı, eldeki imge çiftleri arasında en uygun geometrik dönüşümü bulmaktır. Teknik olarak imge çakıştırma problemi şu şekilde ifade edilebilir; bir referans (sabit imge) ve bir hareketli (bu çalışma için bozulmuş imge) imge verildiğinde, çakıştırılan şablon imgenin referans imgeyle aynı koordinat sistemine taşınması istenir.

İmge çakıştırma işlemi uzaktan algılama, medikal görüntüleme, bilgisayarlı görme gibi farklı alanlarda sıkça kullanılan bir işlemdir. Çakıştırılacak imgelerin elde edilmiş şekline göre 4 farklı grupta toplanabilir [50]:

– **Çoklu Görüş Noktalarından Elde Edilen İmgeler:** Aynı sahnenin imgeleri farklı görüş noktalarından toplanır. Amaç iki boyutlu daha geniş bir görüş alanı veya aynı sahnenin üç boyutlu temsilini elde etmektir. Uzaktan algılama uygulamalarında incelenen alanın imgelerinin birleştirilmesi (mosaic) veya bilgisayarlı görme uygulamalarında şekil dönüşümü gibi amaçlara hizmet edebilir.

– **Çoklu Zamansal İmgeler:** Aynı sahnenin imgeleri görüntüleme sistemi ve konumu değişmeden farklı zamanlarda toplanır. Amaç toplanan ardışık imgelerde sahnede

oluşan değişimleri tespit etmektir. Medikal görüntüleme de tümör gelişimini tespit etmek veya bilgisayarlı görmede otomatik değişiklik tespiti gibi farklı amaçlar için kullanılabilir.

– **Çoklu Sensörlerden Toplanan İmgeler:** Aynı sahnenin imgeleri farklı sensörlerden toplanır. Amaç farklı kaynaklardan toplanan bilgilerin sahnedeki detayları ve karmaşıklığı daha iyi ifade edebilmesidir. Uygulama alanları medikal görüntüleme de manyetik rezonans, ultrason veya bilgisayarlı tomografi gibi anatomik vücut yapılarını görüntüleyen cihazlardan elde edilen imgelerin karşılaştırılıp daha anlamlı bir bilgi elde etmek veya uzaktan algılamada daha iyi uzamsal çözünürlük sağlayan pankromatik imgeler ile yüksek spektral çözünürlüğe sahip multispektral görüntülerinin karşılaştırılması olabilir.

– **Sahneden Modele Çakıştırma:** Bir sahnenin imgesi ve o sahnenin modeli karşılaştırılır. Model sahnenin bilgisayarlı temsili olabilir. Örnek olarak coğrafi bilgi sistemleri için sayısal modeller verilebilir.

Çakıştırılacak imgelerin farklılığı ve çeşitli bozulma türleri nedeniyle, tüm çakıştırma işlemleri için geçerli evrensel bir yöntem tasarlamak imkansızdır. Her yöntemde sadece imgeler arasında varsayılan geometrik deformasyon tipi değil aynı zamanda gürültüden dolayı oluşan bozulmalar, çakıştırma işleminin uygunluğu ve uygulamaya bağlı özellikler de dikkate alınmalıdır.

Çakıştırma yöntemlerinin hemen hepsi şu dört adımdan oluşur:

Özellik tespiti: Farklı özellikler (köşeler, çizgi kesişimleri, kenarlar, kontürler, kapalı sınır bölgeleri v.b) elle veya otomatik olarak algılanır. Bu özellikler literatürde kontrol noktaları olarak adlandırılan nokta temsilcileriyle (ayrıt edici noktalar, çizgi sonları, ağırlık merkezleri) temsil edilebilir.

Özellik Eşleştirme: Bu adımda imge dizisindeki imgede algılanan özellikler ile referans imgede saptanan özellikler arasında ilişki kurulur. Bu amaçla, özellikler arasındaki uzamsal ilişkilerle birlikte çeşitli özellik tanımlayıcıları ve benzerlik ölçüleri kullanılır.

Dönüşüm Modeli Kestirimi: Hareketli imgeyi referans imgeyle aynı koordinat düzlemine getirecek haritalama fonksiyonlarının tipi ve parametreleri tahmin edilir.

İmge Dönüşümü: Hareketli imge, haritalama fonksiyonları aracılığıyla dönüştürülür. Tam sayı olmayan koordinatlardaki imge değerleri, uygun bir ara değerlendirme tekniği ile tahmin edilir.

İmge çakıştırmanın amacı, mevcut yapılar arasındaki ortak özellikleri yakalayarak

farklı imgeler arasında en uygun dönüşümü sağlamaktır. En basit hâliyle imge çakıştırma elemanlarını I_r adında bir referans imge ve I_m adında bir hareketli imge olarak şu şekilde tanımlayabiliriz:

$$\begin{aligned} I_r : \Omega_r &\rightarrow \mathbb{R}, & \Omega_r &\subset \mathbb{R}^{n_r} \\ I_m : \Omega_m &\rightarrow \mathbb{R}, & \Omega_m &\subset \mathbb{R}^{n_m} \end{aligned} \quad (3.1)$$

n_r ve n_m sırasıyla referans ve hareketli imgenin uzamsal boyutlarını temsil etmektedir. Çakıştırma işleminin amacı, tasarlanan bir amaç fonksiyonuna göre, en uygun dönüşüm $T : \Omega_r \rightarrow \Omega_m$ 'yi aramaktır. Böylece, dönüştürülen imge $I_m \circ T$ olarak ifade edilir. Burada \circ bileşke operatörüdür. Herhangi bir $\mathbf{x} = (x_1, x_2, \dots, x_{n_r})$ noktasında $\mathbf{u}(\mathbf{x})$ yer değiştirme vektörü olmak üzere dönüşüm şu şekilde tanımlanabilir:

$$T(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x}). \quad (3.2)$$

İmge çakıştırma işlemi genel olarak üç bölümden oluşur.

1) Dönüşüm Modeli: Hareketli imgeyi referans imgeye eşleyen geometrik dönüşümü tanımlar.

2) Amaç Fonksiyonu: Bir benzerlik ölçütü ve muhtemel bir düzenleştirici terimden oluşur.

İmge çakıştırma, amaç fonksiyonuna göre bir enerji enküçültme sorunu olarak ifade edilebilir:

$$-B(I_r, I_m \circ T) + \varphi R(T). \quad (3.3)$$

Amaç fonksiyonu iki terimden oluşur.

•**Benzeşim Metriği** (B): Dönüştürülen imgenin referans imgeye ne kadar benzediğini ölçer.

•**Düzenleştirici Terimi** (R): Bozulmanın belirli özelliklerini sağlamaya yönelik (örneğin pürüzsüzlük) dönüşümü sınırlandırır ve φ olarak adlandırılan bir sabit tarafından ağırlıklandırılır.

3) Optimizasyon: İmgelerin en uygun şekilde çakıştırılabilmesi için amaç fonksiyonunu minimize eder.

3.1 Dönüşüm Modeli

İmge çakıştırma, bir sahnenin farklı iki imgesi arasında pikselden piksele bir karşılık belirlemek adına, imgelerde tamamlayıcı bilgileri birleştirmek ve imgelerin geometrik farklarının kestirimi için uygulanan hesaplamalı bir işlemdir. Bu işlem genellikle iki imgede de karşılık gelen bir dizi kontrol noktasını otomatik veya manuel belirleyerek bu karşılıklardan imgelerde kalan diğer noktalar için uygun bir dönüşüm modeli belirlemeyi hedefler. Dönüşüm modeli, hareketli imgeyi referans imgeye eşleyecek dönüşümü sağlayacak bozulmaların düzeyini belirler [51]. En basit dönüşüm modelleri gerçek bozulmaların dikkate alınmadığı, sadece genel bir dönme ve öteleme işleminden ibaret olan katı imge çakıştırmada kullanılır. Doğrusal olmayan deformasyonlar mevcutsa daha genel ve karmaşık dönüşüm modelleri dikkate alınır. Dönüşüm modelindeki serbestlik derecesi sayısı, optimizasyonda kullanılacak parametre sayısına eşittir. Daha fazla serbestlik derecesi deformasyonların daha belirgin bir şekilde tanımlanmasını sağlar. Öte yandan bu işlem, aşırı deformasyonlara da neden olabilir, bundan dolayı modelin karmaşıklığı artacaktır. Böylece daha yüksek hesaplama maliyeti ve daha zorlu optimizasyon problemi ortaya çıkacaktır. Örneğin, küçültücünün yüksek boyutlu bir arama alanında yerel bir minimuma ulaşması daha olasıdır.

Katı olmayan imge çakıştırma yöntemlerinde dönüşüm modelleri iki kategoriye ayrılabilir [52]:

- i) fiziksel modeller
- ii) radyal tabanlı fonksiyon gösterimi

Genel olarak fiziksel modeller süreklilik mekaniği teorisinden türetilir. Fonksiyon gösterimleri ise ara değerlendirme ve yaklaşım teorisinden türetilir.

3.1.1 Fiziksel Modeller

Dört alt kategoriye ayrılır: 1) esnek modeller, 2) akışkan modeller, 3) yayılım modelleri, 4) optik-akış modelleri

3.1.1.1 Esnek Modeller

Bu yöntem iki imgeyi eşleştirmek için gereken deformasyonu, kauçuk gibi esnek bir malzemenin deformasyonuna benzer bir fiziksel işlem olarak modeller. Esnek malzemelere uygulanan herhangi bir dış kuvvete, denge durumundan değişime direnen bir iç kuvvetle (malzemenin kendisinin bir özelliği) karşı koyulur. Bu iki

kuvvet eşit olduğunda deformasyon durur [53]. Denge halinde :

$$\chi \nabla^2 \mathbf{u} + (\nu + \chi) \nabla(\nabla \mathbf{u}) + \mathbf{F} = 0, \quad (3.4)$$

Burada \mathbf{F} , esnek gövdeye uygulanan dış kuvveti, \mathbf{u} yer değiştirme vektörünü, χ ve ν ise Lamé'nin elastik sabitlerini temsil etmektedir.

3.1.1.2 Akışkan Modeller

Doğrusal esnek modelinin en önemli şartı, küçük deformasyon varsayımıdır. Akışkan modeller bunun aksine hareketli imgeyi yapışkan bir sıvı olarak temsil ederek bu handikapın üstesinden gelir. Deformasyonlar Navier-Stokes denklemi ile tanımlanmıştır [54]:

$$\chi \nabla^2 \mathbf{v} + (\nu + \chi) \nabla(\nabla \mathbf{v}) + \mathbf{F} = 0, \quad (3.5)$$

Burada da $\mathbf{v} = (x, y, z)$ 3 boyutlu bir sahnede (örneğin ultrason imgelerinde) hız vektörünü (belirli bir bölgedeki hız dağılımı), \mathbf{F} ise modele uygulanan kuvveti temsil etmektedir. Bu denklem esnek modelin denklemine benzeyebilir fakat bu denklem, yer değiştirme vektörü yerine hız vektörünü barındırır.

3.1.1.3 Yayınım Modeli- Demons Yöntemi

Thirion v.d. [55] bozulma işlemini bir yayınım işlemi olarak kabul eden, denklemi aşağıda olan bir yayınım modelini önerdiler:

$$\Delta \mathbf{u} + \mathbf{F} = 0, \quad (3.6)$$

Burada \mathbf{F} , imge eşleştirme ölçütüne dayalı çakıştırmayı yönlendiren bir kuvvettir.

Maxwell'in cini (termodinamiğin 2. yasasının geçerliliğini sorgulamak adına ortaya atılmış bir düşünce deneyi) ilham alan Thirion, çalışmasında imge çakıştırma yöntemi olarak yayınım modelini ilk öneren kişi olmuştur. Temel fikir, hareketli imgedeki bir nesnenin her noktasının, referans imgedeki aynı nesnenin içi veya dışı olarak belirlenebilmesidir. Demons, dış noktaları nesneye itmek ve iç noktaları sınır dışına çekmek için nesne sınırında yer alan etkileyicilerdir. Kuvvetler zaman içinde nesnelerin sabit yoğunluklarını varsayarak, zamansal bir sırada iki imgedeki nesnelere

arasındaki eşleştirmeyi oluşturmak için geliştirilen optik akışa [56] dayanmaktadır. Demons yöntemi, esnek ve akışkan çakıştırma yöntemlerinden çok daha hızlıdır. Pennec [57] çalışmasında katı olmayan çakıştırma işlemini bir optimizasyon problemi olarak araştırmış ve Demons yönteminin bir Gauss yumuşatma düzenleyicisiyle birlikte farkların kareleri toplamı (SSD) olan benzerlik ölçümünün ikinci dereceden bir gradyan inişine eşdeğer olduğunu göstermiştir.

Demons, küçük yer değiştirmeler için geçerli olan optik akış kısıtlaması dikkate alınarak hesaplanır. Düzenlilik, Gauss yumuşatma ile sağlanır.

Bu algoritma iki basamaktan oluşur. Bunlar:

1 : her noktanın (pikselin) yer değişiminin kestirimi

2 : bir düzennileştirme adımı

Optik akış denklemi Demons'u kestirmek için kullanılır. Belirli bir O noktası için, referans imgedeki yeğlilik $I_r(x)$ ve hareketli imgedeki yeğlilik ise $I_m(x)$ olsun. O noktasının hareketli imgede karşılık gelen noktaya uyması için gereken tahmini yer değiştirme [55] tarafından şu şekilde verilmiştir:

$$\mathbf{u} = \frac{(I_m(x) - I_r(x))\nabla I_r(x)}{|\nabla I_r(x)|^2 + (I_m(x) - I_r(x))^2}, \quad (3.7)$$

Burada $\mathbf{u} = (u_x, u_y, u_z)$ ve $\nabla I_r(x)$; referans imgenin gradyanıdır. $\nabla I_r(x)$, referans imgedeki komşu noktalar arasındaki ilişkiyi temsil eder. Bundan dolayı; referans imgeden çıkan bir "içsel" kuvvettir. $(I_m(x) - I_r(x))$ terimi ise, referans ve hareketli imge arasındaki etkileşimin diferansiyel kuvvetidir ve dolayısıyla "dışsal" bir kuvvettir.

Denklem 3.7'de yayınının tek yönlü olduğu varsayılmaktadır. Yayınının çift yönlü olduğu varsayılırsa imge uzayının herhangi bir noktasında Demons sadece, hareketli nesneden referans nesneye yayılmasını sağlayan bir kuvvet değil, ayrıca tersi durumda da bir kuvvet üretir. Böylece [58]'de varsayılan durumla birlikte Demons'a yeni bir kuvvet eklenir.

Denklem 3.7'ye benzeyen diğer bir denklem de hareketli imgenin gradyan bilgisine dayanan \mathbf{F}_{I_m} , aktif kuvveti tanımlamakta kullanılır:

$$\mathbf{u} = -\frac{(I_r(x) - I_m(x))\nabla I_m(x)}{|\nabla I_m(x)|^2 + (I_r(x) - I_m(x))^2}, \quad (3.8)$$

Burada '-' işareti karşı gücü temsil eder. Pasif kuvvet \mathbf{F}_{I_r} ile birlikte O noktasındaki

toplam yer deęiřtirme řu řekilde hesaplanır:

$$F = (I_m(x) - I_r(x)) \cdot \left(\frac{\nabla I_r(x)}{|\nabla I_r(x)|^2 + (I_m(x) - I_r(x))^2} + \frac{\nabla I_m(x)}{|\nabla I_m(x)|^2 + (I_r(x) - I_m(x))^2} \right). \quad (3.9)$$

3.1.1.4 Optik Akıř Modeli - SIFT-Akıř Yöntemi

SIFT-Akıř algoritmasının anlaşılabilmesi için, ölçeklemeden baęımsız özellik dönüşümü veya SIFT (scale invariant feature transform) hakkında genel bilgiye sahip olmak gerekmektedir. SIFT ilk olarak 1999 yılında [59] çalışmasında tanıtılmıştır. Yazarlar burada, özellik çıkarma ve algılama yeteneklerine sahip seyrek bir özellik temsili önermektedir. SIFT algoritmasının birkaç aşaması vardır: İlk aşama bir ölçek uzayı oluşturmaktır. Bu işlem, orijinal imgenin kademeli olarak Gauss yumuřatma operatörüyle yumuřatılmış imgeleri üretilerek yapılır (aynı boyuttaki bulanık imgelerin sayısı ölçek olarak adlandırılır). Daha sonra orijinal imge boyut olarak ikiye bölünür ve ikiye bölünmüş imgenin yumuřatılmış versiyonları aşamalı olarak oluşturulur ve bu işlem istenildięi kadar tekrarlanır. Böylece bir ölçek uzayı oluşturulur.

Bir sonraki aşamada, bütün bulanık imgeler kullanılarak Gauss'ların Farkı (Difference of Gaussians-DoG) ile farklı bir imge seti oluşturulur. Bu işlem Gauss piramidinin içindeki farklı ölçekler için yapılır. Daha sonra bu imgeler ölçek baęımsız olan Gaussların Laplasyanının hesabı için kullanılır. Bir imgede bulunan bir noktanın 3x3 komşuluęunda 8 komşusuyla ve aynı zamanda hem bir önceki ölçek hem de bir sonraki ölçekle karşılaştırması toplam 26 adet denetleme ile yapılır. Eęer bu nokta yerel bir uç nokta ise, potansiyel bir kilit noktasıdır ve bu da o ölçek için en iyi nokta olduęunu gösterir. Bu noktalardan bazıları düşük kontrast deęerine, bazıları da kenarlar içinde bulunur. Bu iki senaryoda da özellikler için uygun bulunmazlar. Bu nedenle bu noktaların elenmesi gerekir. Düşük kontrast deęerleri için bu noktaların yeęinlikleri kontrol edilir. Olası uç noktalar (minimum ve maksimum noktaları) veri olarak kullanılarak Taylor açılımı hesaplanır ve gerçek uç nokta düşük kontrasta sahip deęilse veya bir kenara yayılmamışsa kilit nokta olarak kabul edilir ve etiketlenir. İzleyen aşamada, her bir kilit noktanın gradyan yönelimlerinden bir yönelim histogramı şekillendirilir. Histogramdaki tepe deęerinin yeri saptanır. Bu tepeye karşılık gelen yönelimler ve bu tepenin %80'ine kadar olan kısımlara kilit noktalar yerleştirilir. Son aşamada her bir kilit noktanın bir konumu, ölçeęi ve yönü vardır. Görüş açısı ve aydınlatmadaki deęişimlere karşı mümkün olduęunca deęişmez her bir kilit nokta hakkında yerel imge bölgeleri için bir tanımlayıcı hesaplanır. Bunun için bir kilit nokta etrafında 16x16 komşuluk alınır. Bu komşuluk 4x4 boyutunda alt bloklara bölünür.

Her alt blok için 8 adet yönelim histogramı oluşturulur. Böylece $4 \times 4 \times 8 = 128$ adet yönelim değeri bulunur. Bu sonuç kilit nokta tanımlayıcısının özellik vektörü olarak tanımlanır.

SIFT akış algoritması için yalnızca özellik çıkarma bileşeni - yukarıda belirtilen son aşama - gereklidir. SIFT tanımlayıcı; algılanan kilit noktalar için değil, imgenin her bir pikseli için bulunur. Pikselin 16 piksellik bir komşuluğu, 4×4 hücre dizisine bölünür ve daha sonra her bir hücre dizisi, 8 farklı yönelim histogramıyla temsil edilir. Böylece her piksel için $4 \times 4 \times 8 = 128$ boyutunda (128B) bir özellik vektörü oluşturulur. Daha sonra SIFT tanımlayıcılarını ilgili piksel konumlarına göre sıralayarak bir SIFT imgesi oluşturulur [60].

SIFT-Akış algoritması için, optik akış algoritmasından esinlenilmiştir. Optik akışta, piksellerin yeğinliği eşleştirilir. Bir imgede, bir pikselin en iyi eşleşmesini bulmak için, karşılık gelen imge piksel piksel çaprazlanır ve amaç fonksiyonunu en aza indiren ikinci imgenin pikselinin en iyi eşleşme olduğu düşünülür. SIFT-Akış'ta da aynı işlem yapılır, ancak piksel yeğnilikleri yerine SIFT tanımlayıcıları işlenir.

Yoğunluk Uygunluğu:

Bu yöntemde; karşılaştırılmak istenen iki imgeyi eşleştirmek için, optik akışın işlevine benzer bir işlev kullanılır. Amaç, SIFT tanımlayıcılarının akış vektörünü tahmin etmektir. Bu amaç için iki kriter belirlenir: ilk olarak, SIFT tanımlayıcıları akış vektörleri boyunca eşleştirilmelidir, ikinci olarak akış vektörlerinin yumuşak olması ve sadece kendi sınırları etrafında bozulmuş olması beklenir. Bu kriterlere göre fonksiyon şu enerji denklemiyle hesaplanır:

$$E(\mathbf{w}) = \begin{cases} \sum_{\mathbf{x}} \min(\|I_r(\mathbf{x}) - I_m(\mathbf{x} + \mathbf{w}(\mathbf{x}))\|_1, t) + \\ \sum_{\mathbf{x}} \zeta(|u(\mathbf{x})| + |v(\mathbf{x})|) + \\ \sum_{(\mathbf{x}, \mathbf{q}) \in \epsilon} \min(\alpha|u(\mathbf{x}) - u(\mathbf{q})|, d) + \min(\alpha|v(\mathbf{x}) - v(\mathbf{q})|, d) \end{cases}, \quad (3.10)$$

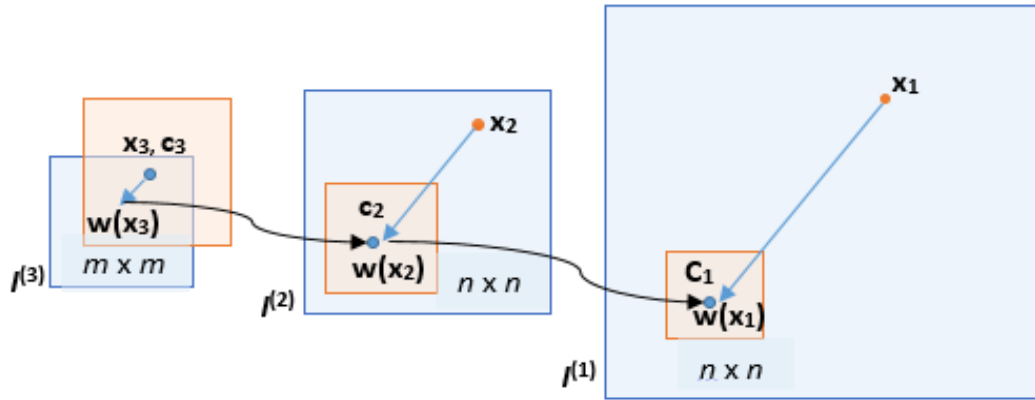
Bu denklemde $\mathbf{x} = (x, y)$ verilen bir imgenin koordinatlarını, $\mathbf{w}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$ \mathbf{x} 'deki akış vektörünü, $u(\mathbf{x})$ ve $v(\mathbf{x})$ de sırasıyla x ve y yönlerindeki bileşenleri, I_r ve I_m ise eşleştirilmek istenen iki SIFT imgeyi temsil etmektedir. $u(\mathbf{x})$ ve $v(\mathbf{x})$ için \mathbb{L} adet durum sözkonusudur. Birinci ve üçüncü terimde kullanılan t ve d ; L_1 normu için eşik değerleridir ve sırasıyla eşleşen aykırı değerlerde sağlamlığı artırmak ve akış alanındaki süreksizlikleri önlemek için kullanılır. ζ ve α , amaç fonksiyonunun sırasıyla ikinci ve üçüncü terimlerini kontrol etmek için kullanılan hiper parametrelerdir.

Denklem 3.10'da ilk terim, akış alanına göre iki SIFT imgenin arasındaki farkın en aza indirilmesinden sorumlu olan veri terimidir. İkinci terime küçük yer değiştirme terimi denir ve diğer bilgiler yetersiz olduğunda akış vektörlerini olabildiğince küçük kalmaya zorlar. Akış vektörlerini pürüzsüz olmaya zorlayan son terim veya pürüzsüzlük terimi, komşu piksellerin akış vektörlerini benzer olmaya zorlar.

Amaç fonksiyonunu optimize etmek için temel algoritma olarak çift katmanlı döngüsel inanç yayılımı kullanılmıştır. Optik akışın genel formülasyonundan farkı, yukarıdaki denklemde pürüzsüzlük terimi yatay ve dikey akışların ayrılmasına izin verecek şekilde ayrıştırılmıştır. Böylece algoritmanın karmaşıklığı $O(L^4)$ 'ten $O(L^2)$ 'ye azaltılmıştır.

Kabandan İnceye Eşleştirme:

SIFT-Akış yönteminin bir diğer önemli bölümü hızıdır. Her ne kadar amaç fonksiyonunun optimizasyonu için çift katmanlı bir döngüsel inanç yayılımı (belief propagation) kullanmanın diğer optimizasyon yöntemlerine kıyasla yakınsama hızını arttırdığı kanıtlanmış olsa da, imge gibi büyük verilerle uğraşırken etkili olduğu söylenemez. Bu sorunu çözmek için Şekil 3.1'de görülen kabandan inceye eşleştirme algoritması geliştirilmiştir. Bu algoritma SIFT akış yönteminin performansını önemli ölçüde hızlandırmıştır.



Şekil 3.1 SIFT-Akış yönteminde kabandan inceye eşleştirme algoritması [60]

Bu algoritmanın ana fikri önce imgenin akış vektörlerini kabaca kestirmek ardından akış vektörlerini kabandan inceye doğru kademeli olarak ilerletmek ve iyileştirmektir. Algoritmada basitçe, I_r referans imge ve I_m hareketli imge I olarak belirtilmiştir. $I^{(1)} = I$ olacak şekilde bir SIFT piramidi $I^{(e)}$ kurulur. Bu piramitte $I^{(e+1)}$, $I^{(e)}$ 'den türetilerek yumuşatılmış ve indirgenmiş olsun. Piramidin her e seviyesinde x_e ; eşleşecek pikselin konumu, c_e ; arama penceresinin merkezi ve $w(x_e)$ ise inanç yayılımının en iyi eşleşmesi olsun. Piramidin en üst seviyesi olan $I^{(3)}$ 'te m ; $I^{(3)}$ 'ün

yüksekliği (genişliği) olmak üzere boyutu $m \times m$ olan bir arama penceresi x_3 'te ($c_3 = x_3$) ortalanır. İnanç yayılımı algoritmasının bu seviyesinde karmaşıklık $O(m^4)$ 'tür. İnanç yayılımı yakınsaklaştıktan sonra, sistem optimize edilmiş akış vektörü $w(x_3)$ 'ü bir sonraki seviye c_2 olacak şekilde ilerletir. Bu arama penceresi $n = 11$ olacak şekilde $n \times n$ boyutundadır. Bu işlem $I^{(3)}$ 'ten $I^{(1)}$ 'e akış vektörü $w(x_1)$ kestirilene kadar yinelenir.

3.1.2 Radyal Bazlı Fonksiyon Gösterimi- İnce Tabaka Kama Yöntemi

İnce tabaka kama yöntemi, bir dizi kontrol noktası arasında uygun bir ara değerlendirme sağlayan tekniklerden biridir. Her kontrol noktasından geçen bir yüzeyin ara değerlemesini yapar. Kontrol noktaları, bükülen bir yüzey üzerindeki yer sınırlandırmaları olarak düşünülebilir. Bu yöntem referans ve hareketli imgeler arasında karşılık gelen iki kontrol noktası setine göre bir deformasyon fonksiyonu belirlemek için kullanılabilir [61]. Bu dönüşüm modeli, doğrusal olmayan ara değerleri hesaplamak için etkili bir yöntemdir.

Bu yöntemde iki koordinat dizi bilgisi seti x_{r_i} ve x_{m_i} olarak tanımlanır. Burada $i = 1, 2, \dots, n$ olmak üzere $x_{r_i} = (x_{r_i}, y_{r_i})$ ve $x_{m_i} = (x_{m_i}, y_{m_i})$ sırasıyla referans ve hareketli imgeye ait kontrol noktalarıdır. Dönüşüm modeli aşağıdaki denklemde olduğu gibidir:

$$u(x_{r_i}) = a_0 + a_1 x_{r_i}^1 + a_2 x_{r_i}^2 + \sum_{i=1}^n \omega_i U(|x_{r_j} - x_{r_i}|), \quad (3.11)$$

Burada $x = (x, y)$ imgelerin koordinatlarıdır. İlk üç terim tüm kontrol noktalarına en uygun gelen bir düzlemi tanımlayan doğrusal bir parçayı (afin dönüşüm) ifade eder. Kestirme operatörü W matrisi $(n + 3) \times 2$ boyutundadır ve şu denklemle ifade edilir:

$$W = (\omega_1 \dots \omega_n a_0 a_1 a_2)^T = L^{-1} M, \quad (3.12)$$

$\omega_1, \omega_2, \dots, \omega_n, a_0 a_1$ ve a_2 İTK'da kullanılan birincil katsayılarıdır. W matrisi denklem 3.12'de görüldüğü gibi aynı şekilde L ve M ile de temsil edilebilir. M matrisi $(n + 3) \times 2$ boyutundadır ve aşağıdaki şekilde tanımlanır:

$$M = (x_{m_1} x_{m_2} \dots x_{m_n} 000)^T, \quad (3.13)$$

Burada $x_{m_i} = (x_{m_i}, y_{m_i})$, $i = 1, \dots, n$ hareketli imgedeki seçili kontrol noktalarının koordinatlarıdır. L matrisi $(n + 3) \times (n + 3)$ boyutundadır ve K matrisi, Q matrisi ve

(3x3)lük bir sıfır matrisiyle şu denklemlerle ifade edilir:

$$L = \begin{bmatrix} K & Q \\ Q^T & 0 \end{bmatrix}, \quad (3.14)$$

L matrisini tanımlarken kullanılan K ve Q matrisi de şu şekildedir:

$$K = \begin{bmatrix} 0 & U_{r_{12}} & \dots & U_{r_{1n}} \\ U_{r_{21}} & 0 & \dots & U_{r_{2n}} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ U_{r_{n1}} & U_{r_{n2}} & \dots & 0 \end{bmatrix}, \quad (3.15)$$

Burada $U_r = r^2 \log r$ ve $r_{ij} = |x_{r_i} - x_{r_j}|$ referans imgedeki iki seçili kontrol noktası arasındaki uzaklıktır. Q matrisi ise şu şekildedir:

$$Q = \begin{bmatrix} 1 & x_{r_1} & y_{r_1} \\ 1 & x_{r_2} & y_{r_2} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & x_{r_n} & y_{r_n} \end{bmatrix}. \quad (3.16)$$

(x_{r_i}, y_{r_i}) , $i = 1, \dots, n$ referans imgenin kontrol noktası koordinatlarını temsil etmektedir.

3.2 Benzerlik Ölçümü

Dönüştürülen imge ile referans imge arasındaki benzerlik derecesini ölçmek için benzerlik ölçümü yapılmalıdır. Bu metrikler özelliğe ve yeğinliğe dayalı olarak ölçülebilir. Özelliğe dayalı benzerlik ölçümleri; göze çarpan noktalar, kenarlar ve yüzeyler gibi özelliklerin çıkarılması için ön bir işlem gerektirir [50]. Buna karşılık, yeğlilik temelli metriklerin böyle bir ön işleme ihtiyacı yoktur ve doğrudan imgelerden hesaplanabilir. Özellik tabanlı benzerlik metrikleri, imgelerin ayırt edici nesnelerin detaylı ayrıntılarına sahip olmasını gerektirir. Buna ek olarak, özellik seçimi uygulamaya bağlı olarak yeğninliğe dayalı metriklerden daha özeldir. Bu tür metriklerin seçimi, uygulamaya ve dönüşüm modeline bağlıdır. Bu seçim özellikle, imgelerdeki

karşılık gelen yeğinlikler arasında beklenen ilişkiye dayanmalıdır.

Optimizasyonun amacı imgeler arasındaki farklılığı en aza indirmektir. Amaç fonksiyonunun enerji minimizasyonu ile tutarlı olması için, imgelerin farkını hesaplayan benzerlik metrikleri doğrudan amaç fonksiyonuna dahil edilir.

3.2.1 İmge Fark Ölçümü

En basit benzerlik ölçümü, referans ile dönüştürülen hareketli imge arasındaki yeğinlik farkının karelerinin toplamını almaktır (SSD-sum of square difference).

$$SSD(I_r, I_m) = \frac{1}{N} \sum_{\mathbf{x} \in \Omega} |I_r(\mathbf{x}) - I_m(T(\mathbf{x}))|^2 \quad (3.17)$$

Burada \mathbf{x} ; pikselin konumu ve $T(\mathbf{x})$; dönüşümden sonraki pikselin konumudur. N ise imge düzlemindeki piksellerin toplam sayısıdır.

3.2.2 İmge İlişkisi

Çapraz İlişkide (CC-cross correlation) imge yeğinlikleri arasında doğrusal bir ilişki olduğu varsayılır.

$$CC(I_r, I_m) = \frac{1}{N} \sum_{\mathbf{x} \in \Omega} I_r(\mathbf{x}) \cdot I_m(T(\mathbf{x})) \quad (3.18)$$

Genelde ilinti katsayısı olarak da bilinen normalize edilmiş çapraz ilişki ise (NCC-normalized cross correlation veya correlation coefficient) imge çakıştırma işlemini değerlendirmek için fazlasıyla kullanılmaktadır [62] [63].

$$NCC(I_r, I_m) = \frac{\sum_{\mathbf{x} \in \Omega} (I_r(\mathbf{x}) - \mu_r)(I_m(T(\mathbf{x})) - \mu_m)}{\sqrt{\sum_{\mathbf{x} \in \Omega} (I_r(\mathbf{x}) - \mu_r)^2} \sqrt{\sum_{\mathbf{x} \in \Omega} (I_m(T(\mathbf{x})) - \mu_m)^2}} \quad (3.19)$$

Burada da μ_r ve μ_m , referans ve hareketli imgenin yeğinlik ortalamalarıdır.

3.2.3 Karşılıklı Bilgi Miktarı

Bilgi teorisinde imge çakıştırma, referans ve hareketli imge arasında paylaşılan bilgi miktarını en üst düzeye çıkarma süreci olarak düşünülebilir. İmge içeriğinin bilgileri

Shannon entropisi ile ölçülebilir [64]:

$$\mathbb{H} = - \sum_{\mathbf{x}} h(I(\mathbf{x})) \log(h(I(\mathbf{x}))) \quad (3.20)$$

Buradaki $h(I(\mathbf{x}))$; $I(\mathbf{x})$ yeğniliğine sahip imgedeki bir pikselin olasılığıdır. Bu olasılık, yeğnilik histogramından kestirilir. Ortak entropi şöyle tanımlanabilir:

$$\mathbb{H}(I_r, I_m) = - \sum_{\mathbf{x}} h(I_r(\mathbf{x}), I_m(\mathbf{x})) \log(h(I_r(\mathbf{x}), I_m(\mathbf{x}))) \quad (3.21)$$

$h(I_r(\mathbf{x}), I_m(\mathbf{x}))$ sırasıyla $I_r(\mathbf{x})$ ve $I_m(\mathbf{x})$ yeğniliklerine sahip referans ve hareketli imgenin ortak olasılık dağılımıdır. Ortak entropinin düşük değerleri normalde imgelerin uygun çakıştırılması ile ilişkilidir. Bununla birlikte, büyük ölçüde imgelerin örtüşmesine de bağlıdır. Örneğin, örtüşme arka planda sadece bir bölge olduğunda, bu yanlış eşleşme aynı zamanda düşük ortak entropi de oluşturur [65]. Karşılıklı bilgi imgelerin marjinal entropilerini içermektedir, böylece örtüşmelere ortak entropiden daha az duyarlı hale gelmektedir. Bu sorunu hafifletmek adına, bu bilgi kullanılabilir ve aşağıdaki denklemlerle ifade edilmektedir:

$$\mathbb{I}(I_r, I_m) = \mathbb{H}(I_r) + \mathbb{H}(I_m) - \mathbb{H}(I_r, I_m) = \sum_{\mathbf{x}} h(I_r(\mathbf{x}), I_m(\mathbf{x})) \log \frac{h(I_r(\mathbf{x}), I_m(\mathbf{x}))}{h(I_r(\mathbf{x}))h(I_m(\mathbf{x}))} \quad (3.22)$$

Karşılıklı bilgi $\mathbb{I}(I_r, I_m)$, I_r 'nin I_m 'yi ne kadar iyi açıkladığını veya tam tersini gösterir. İki imge en iyi şekilde çakıştırıldığında, karşılıklı bilgiler en üst düzeye çıkarılır (amaç fonksiyonu en aza indirilir). Böylece yeni benzerlik ölçümü şu şekilde olur:

$$KB(I_r, I_m) = -\mathbb{I}(I_r, I_m) \quad (3.23)$$

3.3 Düzenlilik Terimi

Düzenleştirme, çözümü zor olan bir problemi kolay çözmek için önemli bir adımdır. Bir problem eğer;

i) çözümü varsa,

ii) çözümü tekse,

iii) girişe göre belirli bir denklem dahilinde süreklirse,

bu problem çözümü kolay bir problemdir. Aksi halde bu tarz problemler çözümü zor

problemlerdir ve buna bağılı bir çok problem çözümü zor olarak adlandırılır. İmge çakıştırma da bu şekilde çözümü zor bir problemdir [66].. Düzenleştirme teorisinin kaynağı Tikhonov'a [67] dayanır. Bu problem amaç fonksiyonuna bir düzenlilik terimi eklenerek biraz daha kolaylaştırılabilir. Bu terim, çakıştırmanın sonucunu kısıtlamak için arka planda bulunan deformasyonların fiziksel özellikleri hakkında önceden bilgi verilmesine izin verir ve enerji fonksiyonu şu şekilde ifade edilir:

$$E(f) = S(f) + \varphi R(f) \quad (3.24)$$

Burada $S(f)$ amaç fonksiyonudur. $R(f)$; düzenleştirici terim, φ ise takas (trade-off) parametresidir. Denklem 3.25'te f fonksiyonunu, x_i konumlarında verilen veri noktaları y_i 'lere uyduran bir düzenleştirme denklemi görülmektedir:

$$E(f) = \sum_i (y_i - f(x_i))^2 + \varphi \|Pf\|^2 \quad (3.25)$$

Burada P doğrusal bir operatördür. Bir operatör sonlu veya sonsuz boyutlu, doğrusal veya doğrusal olmayan bir yapıda olabilir. Çoğu zaman düzenleştirme problemi analitik olarak çözülür. Çözüm Green fonksiyonu kullanılarak ilerler. Bu haliyle denklem şöyle olur:

$$f(x) = \sum_i w_i G(x, x_i) + \psi(x) \quad (3.26)$$

Burada x_i noktalarındaki katsayılar, $w_i = \frac{(y_i - f(x_i))}{\varphi}$ üstteki denklemden gözlenir. $\psi(x)$; değeri sıfır veya bir sabit olan P operatörünün boş uzayıdır.

3.4 Optimizasyon

İmge çakıştırmanın amacı, iki imge arasında olası bir veya daha fazla düzenleyici terimin eklenmesi ile bir benzerlik metriğine dayanan amaç fonksiyonunu en aza indiren optimum dönüşümü aramaktır. Optimizasyon yöntemleri ayrıntılı olarak [68] [69] [70] kaynaklarında ele alınmıştır. Arama alanının doğasına bağılı olarak, optimizasyon yöntemleri

- i) parametrelerin sürekli değerlerde aranması
- ii) parametrelerin ayrık bir kümede aranması

olarak iki tipte sınıflandırılabilir.

3.4.1 Sürekli Optimizasyon

Sürekli optimizasyon yöntemlerinde, değişkenler gerçek değerleri alır ve minimum değere ulaşılan kadar yinelemeli arama ile bir çözüm bulunur. ξ parametreleri şu kurala göre güncellenir:

$$\xi_{k+1} = \xi_k + \gamma_k \delta_k, \quad k = 0, 1, 2, \dots \quad (3.27)$$

Burada γ_k adım boyutunu, bununla birlikte k . döngüdeki kazanç faktörünü temsil etmektedir. Sürekli optimizasyon yöntemleri γ_k adım boyutu ve δ_k arama yönünün hesaplanmasında farklılık gösterir. Adım boyutu sabit olacak şekilde ayarlanabilir, maliyet fonksiyonunu en aza indirmek için tam veya kesin olmayan çizgi arama ile hesaplanabilir. Arama yönü normalde maliyet fonksiyonunun birinci dereceden veya ikinci dereceden türevlerinden hesaplanır.

3.4.1.1 Gradyan İniş

Parametre uzayında maliyet fonksiyonunu azaltmakla görevli olan gradyan iniş metodu, problemler için optimum çözüm üreten genel bir sürekli optimizasyon algoritmasıdır.

İşlem başlangıcında, herhangi bir örnek seçilir ve öğrenme oranına göre her bir basamakta ilerleme sağlanır. Burada öğrenme oranının doğru seçilmesi önemlidir. Eğer olması gerekenden daha büyükse her bir iterasyonda bir öncekinden daha yüksek bir noktaya gidebilir. Olması gerekenden daha küçük seçilmesi ise, minimuma gidene kadar çok fazla işlem yapacağından, öğrenme süresi gereğinden fazla uzun olacaktır.

Gradyan iniş değişkenleri, adım büyüklüğünü hesaplama aşamasında farklılık gösterir. Adım boyutu, altın oran araması tam satır araması [71], kesin olmayan satır araması [72] veya altın oran araması [73] kullanılarak, sabitlenecek şekilde ayarlanabilir.

3.4.1.2 Stokastik Gradyan İniş

Stokastik gradyan iniş yönteminde gradyan değerleri hesaplanırken her bir adımda belirli örnekler değil de rastgele örnekler seçilerek işlem yapılır. Bu yöntem her bir adımda bütün veri setini kullanmadığından, algoritmaya hız katar. Rastgele seçilen örnekler, maliyet fonksiyonunun minimumunda değil de etrafında olmasını sağlar. Bu nedenle, bu algoritma sonunda çıkan parametre sonuçlarının optimal değil de belirlenen kriterlere uygun olduğu bilinmektedir.

3.4.1.3 Toplu Gradyan İniş

Bu algoritmada hesaplama yapılırken her bir adımda bütün veri seti kullanılır. Bu nedenle öğrenme süresi diğer yöntemlerden daha uzun sürmektedir.

3.4.2 Ayrık Optimizasyon

Ayrık optimizasyonda, değişkenlerin ayrık değerleri vardır ve en uygun çözüm bir ayrık kümede aranır. Bu optimizasyon yaklaşımında genellikle Olasılıksal Grafik Modeller (Probabilistic Graphical Models-OGM) ve Markov Rastgele Alanlar (Markov Random Field) kullanılır [74]. OGM, farklı bir etiket kümesinde değerler alan rastgele değişkenleri temsil eden düğümlerden ve değişkenler arasındaki ilişkiyi temsil eden kenarlardan oluşur. Bu yöntemde, etiket kümesi bir dizi olası deformasyonla ilişkilidir. Böylece optimizasyon, amaç fonksiyonunu en aza indirmek için her bir ızgara noktası için en iyi noktayı arar. Böyle bir problemi çözmek için algoritmalar minimum kesme ve inanç yayılımı olarak sınıflandırılabilir.

3.4.2.1 Minimum Kesme

Minimum kesme (min-cut) problemleri genellikle maksimum akış algoritması kullanılarak çözülür. İkili etiketleme probleminde, yönlendirilmiş bir grafikte kaynak ve çıkış adı verilen iki özel düğüm bulunur. Bir kesimin maliyeti, grafiğin kenarlarından geçen toplam akıştır. Bu değer kenarların ağırlıkları ve düğümlerin kapasitesi ile kontrol edilir. Minimum kesimi bulmak, kaynaktan çıkışa akışı maksimuma çıkaran yol aranarak yapılır [75]. [76] çalışmasında araştırmacılar, katı olmayan imge çakıştırma problemi için her pikseli bir düğüm olarak temsil eden yoğun bir yer değiştirme vektörü kullanmışlardır ve amaç fonksiyonunu çizge kesim (graph cut) kullanarak optimize etmişlerdir. Bu algoritmanın en önemli dezavantajı, hesaplama karmaşıklığı nedeniyle düğüm sayısı ve arama alanının sınırlı olmasıdır.

3.4.2.2 İnanç Yayılımı

İnanç yayılımı aynı zamanda mesaj iletme olarak da bilinir. Buradaki olay, her düğümün komşularına en uygun etiket hakkındaki inançlarıyla bir mesaj göndermesidir. Bu komşu düğümler inançlarını günceller ve bir sonraki komşulara mesaj gönderir. Bu süreç tüm düğüm inançları aynı olana kadar tekrarlanır. [77] yayınında araştırmacılar, arama alanını hiyerarşik olarak azaltarak inanç yayılım yönteminin verimliliğini artırmıştır.

3.5 İmge Kalite Değerlendirmesi

Bu üç farklı katı olmayan çakıştırma yöntemleri sonucunda oluşan çıktıları değerlendirmek için tepe sinyal gürültü oranı (PSNR), yapısal benzerlik indis ölçümü (SSIM) ve ilinti katsayısı (NCC) gibi kalite ölçüm yöntemleri kullanılmıştır. Subjektif ölçüm, insan gözü üzerindeki görsel etki ile ilgili olsa da, objektif ölçüm piksellerin değerlerini ölçer.

3.5.1 PSNR

İki imge arasındaki tepe sinyali gürültü oranı desibel olarak hesaplanır. Bu oran, referans ve çakıştırılmış imge arasında bir kalite ölçümü olarak kullanılır. PSNR ne kadar yüksek olursa, çakıştırılmış imgenin kalitesi o kadar iyi olur. Denklemi de şu şekildedir [78]:

$$PSNR = 10 \log_{10} \frac{(2^d - 1)^2 (mn)}{\sum_{x=1}^m \sum_{y=1}^n [I_r(x) - I_c(x)]^2} \quad (3.28)$$

Burada d ; imgenin bit sayısı (8 bit), $[m, n]$ imgenin boyutları ve $I_r(x)$, $I_c(x)$ de sırasıyla referans ve çakıştırılmış imgedeki x .nci satır ve y .nci sütundaki yeğinlikleri ifade etmektedir.

3.5.2 SSIM

Yapısal benzerlik indisi; parlaklık, kontrast ve yapı gibi üç adet bileşen içerip imgenin nesnel kalitesini ölçmektedir. $[-1, 1]$ aralığında sonuç çıkarmaktadır[79]. Çıktı değeri 1'e ne kadar yakınsa karşılaştırılan iki imge birbirine o kadar benzerdir sonucuna varılır. Ters durumda çıktı -1'e ne kadar yakınsa performansın bir o kadar kötü olduğu sonucuna varılır. SSIM ölçümü şu şekildedir:

$$SSIM = \underbrace{\left[\frac{2\mu_{I_r}\mu_{I_c} + C_1}{\mu_{I_r}^2 + \mu_{I_c}^2 + C_1} \right]}_{\text{parlaklık}} \underbrace{\left[\frac{2\Sigma_{I_r}\Sigma_{I_c} + C_2}{\Sigma_{I_r}^2 + \Sigma_{I_c}^2 + C_2} \right]}_{\text{kontrast}} \underbrace{\left[\frac{2\Sigma_{I_r I_c} + C_3}{\Sigma_{I_r}\Sigma_{I_c} + C_3} \right]}_{\text{yapı}} \quad (3.29)$$

Burada μ_{I_r} ve μ_{I_c} , sırasıyla $I_r(x)$ (referans imge) ve $I_c(x)$ (çakıştırılmış imge) imgelerine ait ortalama değerlerdir. $\Sigma_{I_r}^2$, $\Sigma_{I_c}^2$ ve $\Sigma_{I_r I_c}$ ise sırasıyla bu imgelerin varyansları ve kovaryansını temsil etmektedir. C_1 , C_2 ve C_3 de parlaklık, kontrast ve yapı bileşenlerinde paydaların sifira yakınsaması durumunda kararsızlığı engellemek

için kullanılır. Eğer $C_3 = C_2/2$ alınırsa yeni SSIM denklemi şu şekilde ifade edilir:

$$SSIM = \left[\frac{(2\mu_{I_r}\mu_{I_c} + C_1)(2\Sigma_{I_r I_c} + C_2)}{(\mu_{I_r}^2 + \mu_{I_c}^2 + C_1)(\Sigma_{I_r}^2 + \Sigma_{I_c}^2 + C_2)} \right] \quad (3.30)$$

3.5.3 NCC

İlinti katsayısı (NCC); $[0,1]$ arasında bir çıktı verir ve iki imge arasındaki benzerliği tanımlamaya yarayan nesnel bir kalite ölçümüdür. Çıktı 1'e ne kadar yakınsa karşılaştırma o kadar iyi demektir. Formülü ise şu şekilde verilir.

$$NCC = \frac{\sum[(I_r(x) - \mu_{I_r})(I_c(x) - \mu_{I_c})]}{\sqrt{\sum(I_r(x) - \mu_{I_r})^2} \sqrt{\sum(I_c(x) - \mu_{I_c})^2}} \quad (3.31)$$

Burada da μ_{I_r} ve μ_{I_c} ; referans ve karşılaştırılmış imgenin yeğinlik ortalamalarıdır.

3.6 Deneysel Sonuçlar

Bu bölümde atmosferik türbülansın dolaylı olarak oluşan bozulmalara sahip veri setleri üzerinde en etkin katı olmayan imge karşılaştırma yöntemlerinden üçü olan Demons yöntemi, İTK yöntemi ve SIFT-Akış yöntemleri uygulanmış ve sonuçları gösterilmiştir. Şekil 3.2'den Şekil 3.13'e kadar olan imgeler, yöntemlerin görsel karşılaştırmalarını içermektedir. Referans imge, karşılaştırma işlemine sokulan 20 adet bozulmuş imgenin zamansal ortalamasının alınmasıyla elde edilmiştir.

Şekil 3.2'den Şekil 3.13'e kadar olan şekillerde 1. satırda sol taraftaki imge bozulmuş ham imgeyi, sağ taraftaki imge ise bütün yöntemler için kullanılan referans imgeyi temsil etmektedir. 2., 3. ve 4. satırdaki şekiller ise sırasıyla Demons, İTK ve SIFT-Akış yöntemlerinin sonuçlarını göstermektedir. Burada sol taraftaki imgeler sırasıyla bu yöntemler sonucu çıkan karşılaştırılmış imgeyi, ortadaki şekiller o yöntemin o imge için bozulma haritasını ve sağ taraftaki imgeler ise bozulmuş imge ve karşılaştırılmış imge arasındaki farkın görsel şeklidir. Bozulma haritası oluşturulurken her ızgara arası 40 piksel olacak şekilde ayarlama yapılmıştır.

Tablo 3.1, Tablo 3.2 ve Tablo 3.3'te sırasıyla Demons yöntemi, İTK yöntemi ve SIFT-Akış yöntemleri için nesnel metrik sonuçları ve her bir yöntemin işlem süresi görülmektedir. Bütün veri setleri için performans iyileştirmesi açısından en etkili yöntem Tablo 3.3'ten de görüldüğü üzere SIFT-Akış yöntemidir. Fakat bu karşılaştırma yönteminde daha karmaşık hesaplamalar bulunduğundan işlem süresi diğer iki yöntemden daha uzun

Tablo 3.1 Demon's Yöntemi için Kalite Metrik Sonuçları ve İşlem Süresi

| İmge Seti | Demon's Yöntemi | | | |
|--------------|-----------------|--------|--------|--------------|
| | PSNR | SSIM | NCC | İşlem Süresi |
| Veri Seti-1 | 35.62 | 0.9527 | 0.9832 | 2dk.21sn. |
| Veri Seti-2 | 28.09 | 0.8664 | 0.9636 | 1dk.50sn. |
| Veri Seti-3 | 28.16 | 0.8807 | 0.9502 | 1dk.58sn. |
| Veri Seti-4 | 23.65 | 0.8011 | 0.8869 | 2dk.10sn. |
| Veri Seti-7 | 27.22 | 0.8854 | 0.9610 | 1dk.10sn. |
| Veri Seti-8 | 32.84 | 0.9314 | 0.9758 | 1dk.32sn. |
| Veri Seti-9 | 25.57 | 0.8281 | 0.9199 | 1dk.44sn. |
| Veri Seti-13 | 30.40 | 0.8973 | 0.9708 | 1dk.56sn. |
| Veri Seti-14 | 26.78 | 0.8833 | 0.9351 | 2dk.54sn. |
| Veri Seti-19 | 21.89 | 0.8038 | 0.9231 | 2dk.6sn. |
| Veri Seti-20 | 22.07 | 0.7716 | 0.9179 | 2dk.10sn. |
| door [3] | 23.32 | 0.7423 | 0.9025 | 2dk.1sn. |

sürmüştür. İşlem maliyeti en düşük olan yöntem İTK yöntemi olup, performans iyileştirmesi en az olan buna rağmen işlem süresi en kısa olan yöntemdir.

Bütün algoritmalar, MATLAB R2015b programı kullanılarak aynı veri setlerinden 20 adet bozulmamış imgeye uygulanmıştır.

Şekil 3.2'de veri seti-1 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %9.1, İTK'ya göre ise %42.13'tür. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %19.05, İTK'ya göre ise %75.89'dur.

Şekil 3.3'de veri seti-2 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %11.08, İTK'ya göre ise %22.25'tir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %14.84, İTK'ya göre ise %27.74'tür.

Şekil 3.4'de veri seti-3 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %10.48, İTK'ya göre ise %28.37'dir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %14.46, İTK'ya göre ise %40.5'tir.

Şekil 3.5'de veri seti-4 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %12.82, İTK'ya göre ise %19.91'dir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %29.80, İTK'ya göre ise %31.61'dir.

Şekil 3.6'da veri seti-7 için SSIM metriğine göre SIFT-Akış yönteminin Demons

Tablo 3.2 İTK Yöntemi için Kalite Metrik Sonuçları ve İşlem Süresi

| İmge Seti | İTK Yöntemi | | | |
|--------------|-------------|--------|--------|--------------|
| | PSNR | SSIM | NCC | İşlem Süresi |
| Veri Seti-1 | 30.65 | 0.9257 | 0.9436 | 2dk.4sn. |
| Veri Seti-2 | 27.50 | 0.8472 | 0.9571 | 1dk.40sn. |
| Veri Seti-3 | 26.96 | 0.8509 | 0.9284 | 1dk.48sn. |
| Veri Seti-4 | 23.51 | 0.7835 | 0.8839 | 2dk.08sn. |
| Veri Seti-7 | 26.96 | 0.8853 | 0.9585 | 1dk.08sn. |
| Veri Seti-8 | 31.91 | 0.9260 | 0.9692 | 1dk.22sn. |
| Veri Seti-9 | 25.38 | 0.8104 | 0.9157 | 1dk.22sn. |
| Veri Seti-13 | 29.93 | 0.8820 | 0.9673 | 1dk.46sn. |
| Veri Seti-14 | 26.22 | 0.8690 | 0.9230 | 2dk.46sn. |
| Veri Seti-19 | 22.28 | 0.8146 | 0.9339 | 2dk.02sn. |
| Veri Seti-20 | 22.23 | 0.7809 | 0.9250 | 1dk.58sn. |
| door [3] | 22.11 | 0.6842 | 0.8654 | 1dk.48sn. |

Tablo 3.3 SIFT-Akış Yöntemi için Kalite Metrik Sonuçları ve İşlem Süresi

| İmge Seti | SIFT-Akış Yöntemi | | | |
|--------------|-------------------|--------|--------|--------------|
| | PSNR | SSIM | NCC | İşlem Süresi |
| Veri Seti-1 | 36.36 | 0.9570 | 0.9864 | 4dk.25sn. |
| Veri Seti-2 | 28.67 | 0.8812 | 0.9690 | 2dk.48sn. |
| Veri Seti-3 | 28.60 | 0.8932 | 0.9574 | 2dk.40sn. |
| Veri Seti-4 | 24.40 | 0.8266 | 0.9206 | 2dk.36sn. |
| Veri Seti-7 | 27.92 | 0.9028 | 0.9696 | 1dk.18sn. |
| Veri Seti-8 | 33.11 | 0.9378 | 0.9778 | 1dk.33sn. |
| Veri Seti-9 | 25.95 | 0.8428 | 0.9290 | 1dk.55sn. |
| Veri Seti-13 | 30.83 | 0.9050 | 0.9744 | 2dk.46sn. |
| Veri Seti-14 | 27.09 | 0.8933 | 0.9415 | 4dk.42sn. |
| Veri Seti-19 | 22.53 | 0.8266 | 0.9392 | 2dk.33sn. |
| Veri Seti-20 | 22.76 | 0.8045 | 0.9409 | 2dk.48sn. |
| door [3] | 24.43 | 0.7842 | 0.9286 | 2dk.43sn. |

yöntemine karşı performans iyileştirmesi %15.18, İTK'ya göre ise %15.26'dır. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %22.05, İTK'ya göre ise %26.75'tir.

Şekil 3.7'de veri seti-8 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %9.33, İTK'ya göre ise %15.95'tir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %8.26, İTK'ya göre ise %27.92'dir.

Şekil 3.8'de veri seti-9 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %8.55, İTK'ya göre ise %17.09'dur. NCC

metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %11.36, İTK'ya göre ise %15.78'dir.

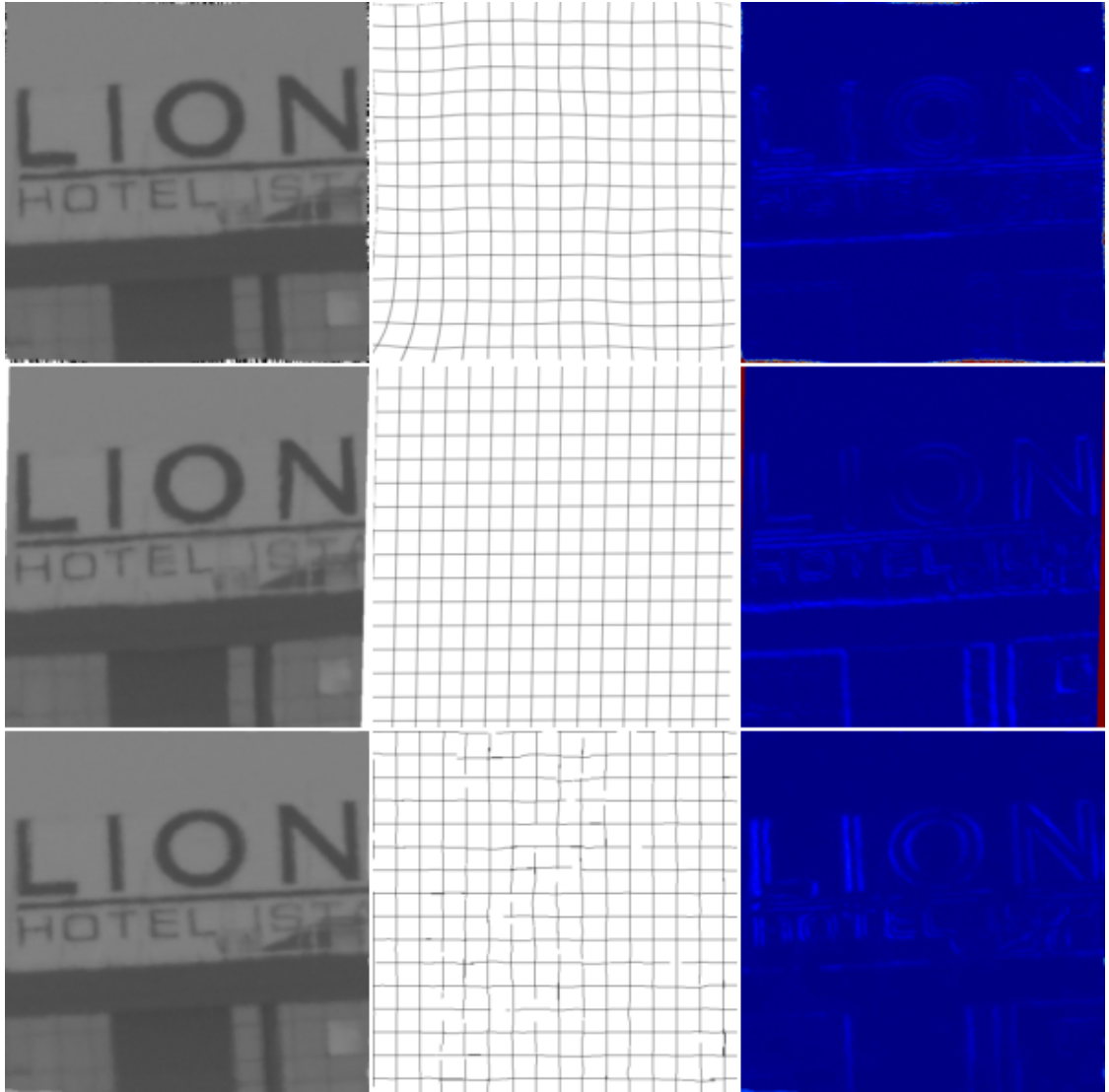
Şekil 3.9'da veri seti-13 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %7.5, İTK'ya göre ise %19.49'dur. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %12.33, İTK'ya göre ise %21.71'dir.

Şekil 3.10'da veri seti-14 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %8.57, İTK'ya göre ise %18.55'tir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %9.86, İTK'ya göre ise %24.03'tür.

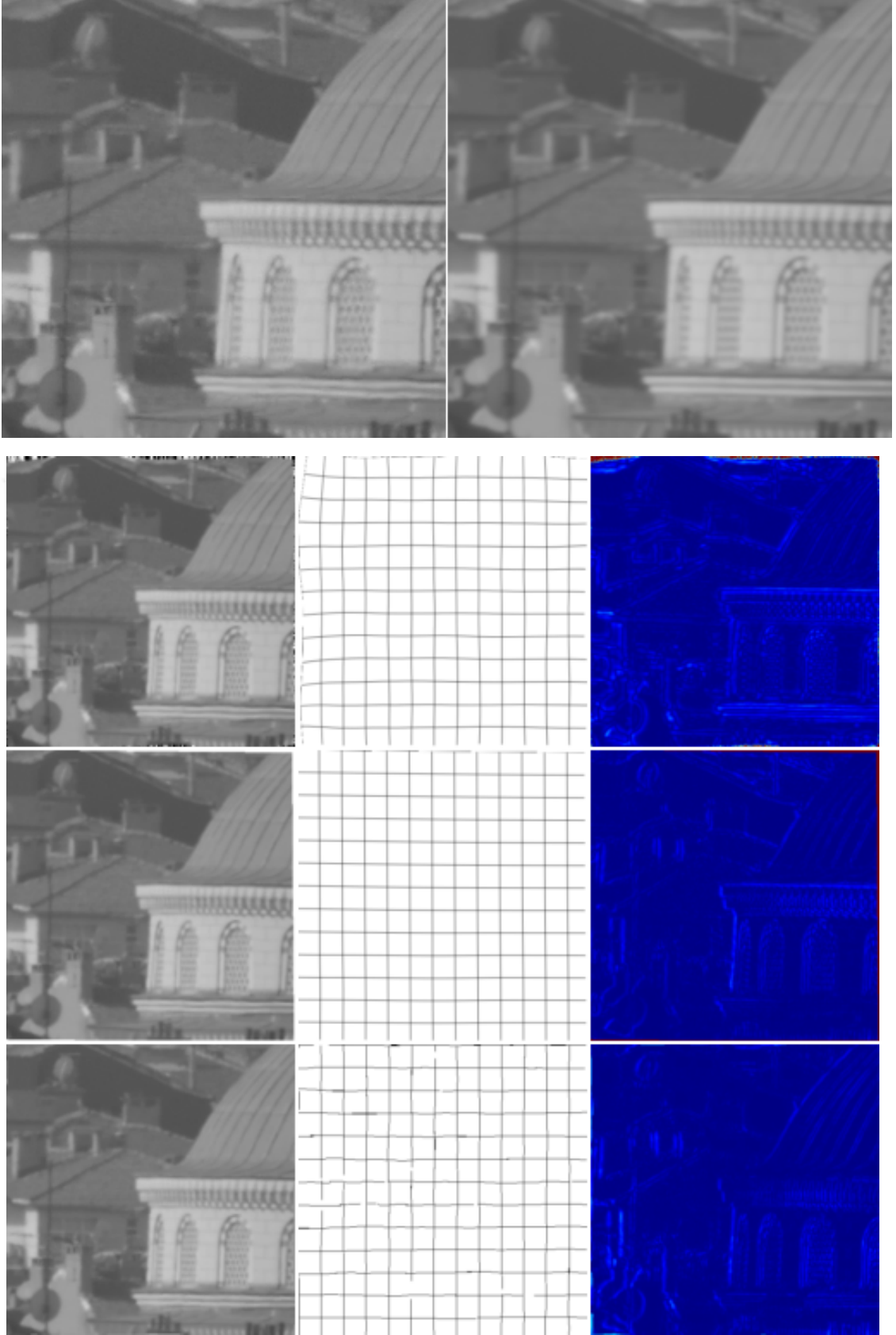
Şekil 3.11'de veri seti-19 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %11.62, İTK'ya göre ise %6.47'dir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %20.94, İTK'ya göre ise %8.02'dir.

Şekil 3.12'de veri seti-20 için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %14.4, İTK'ya göre ise %10.77'dir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %28.01, İTK'ya göre ise %21.20'dir.

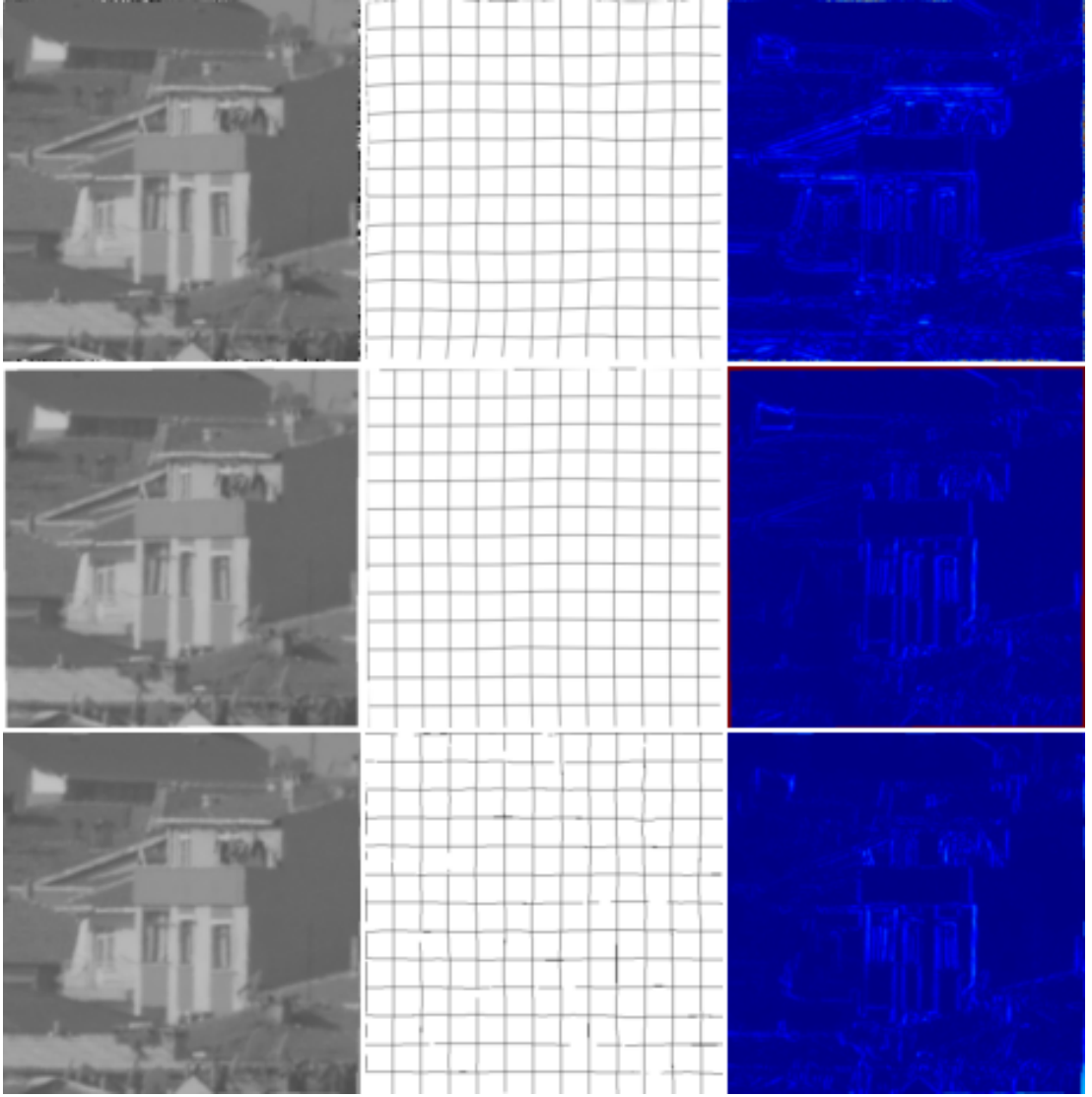
Şekil 3.13'de door[3] veri seti için SSIM metriğine göre SIFT-Akış yönteminin Demons yöntemine karşı performans iyileştirmesi %16.26, İTK'ya göre ise %31.67'dir. NCC metriğine göre ise performans iyileştirmesi oranı Demons yöntemine göre %26.77, İTK'ya göre ise %46.95'tir.



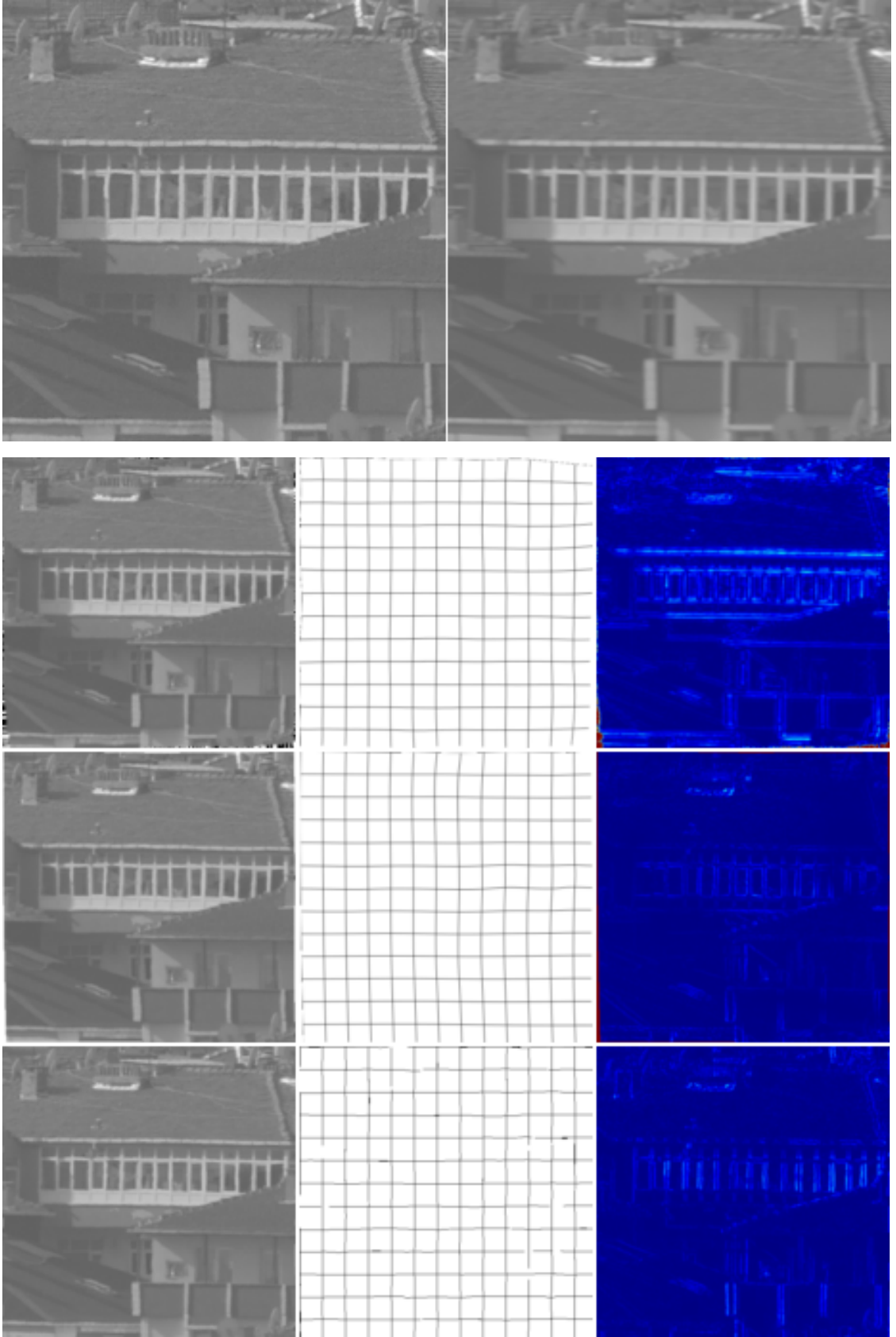
Şekil 3.2 Veri seti-1 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



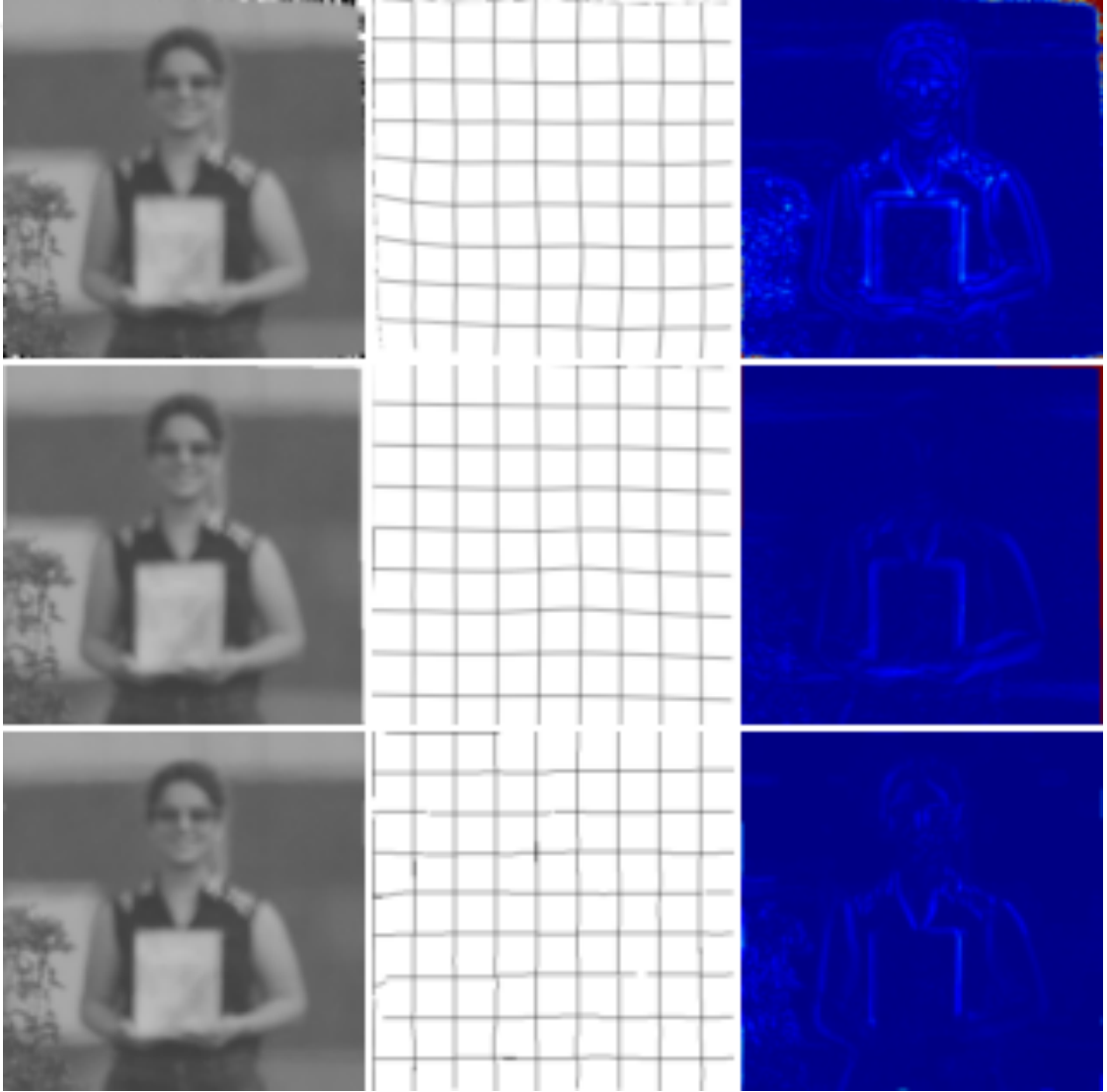
Şekil 3.3 Veri seti-2 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



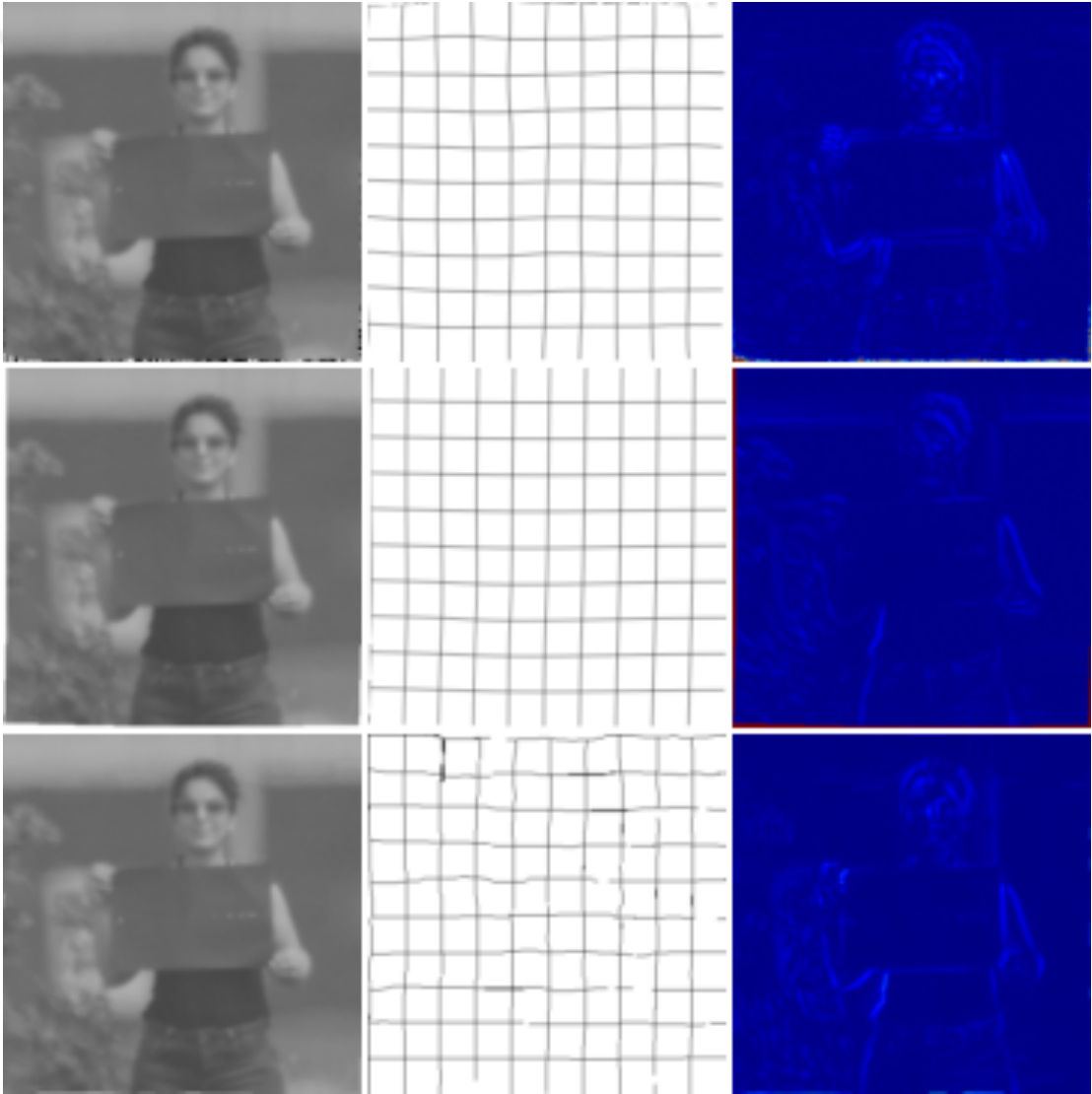
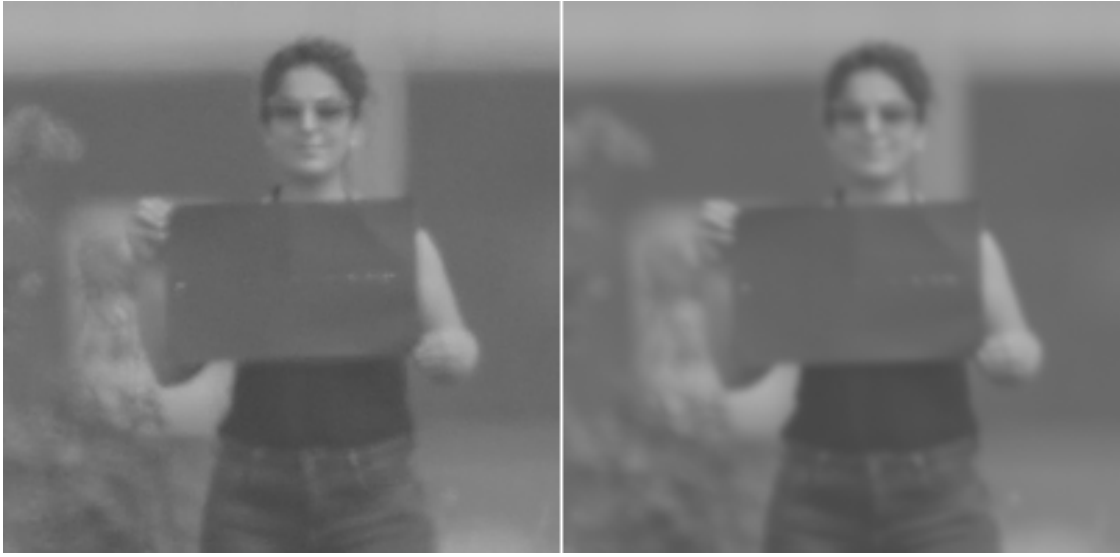
Şekil 3.4 Veri seti-3 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



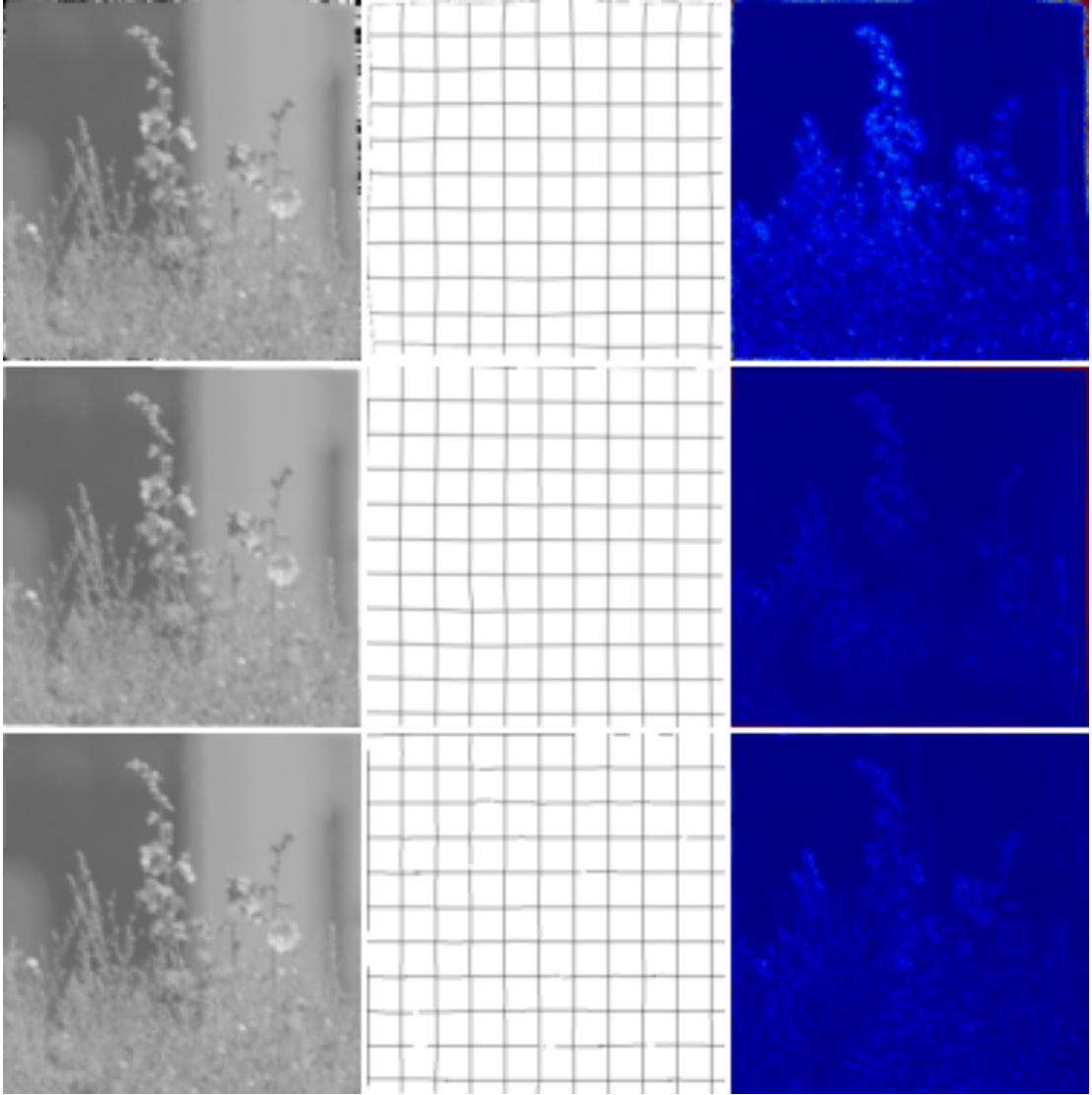
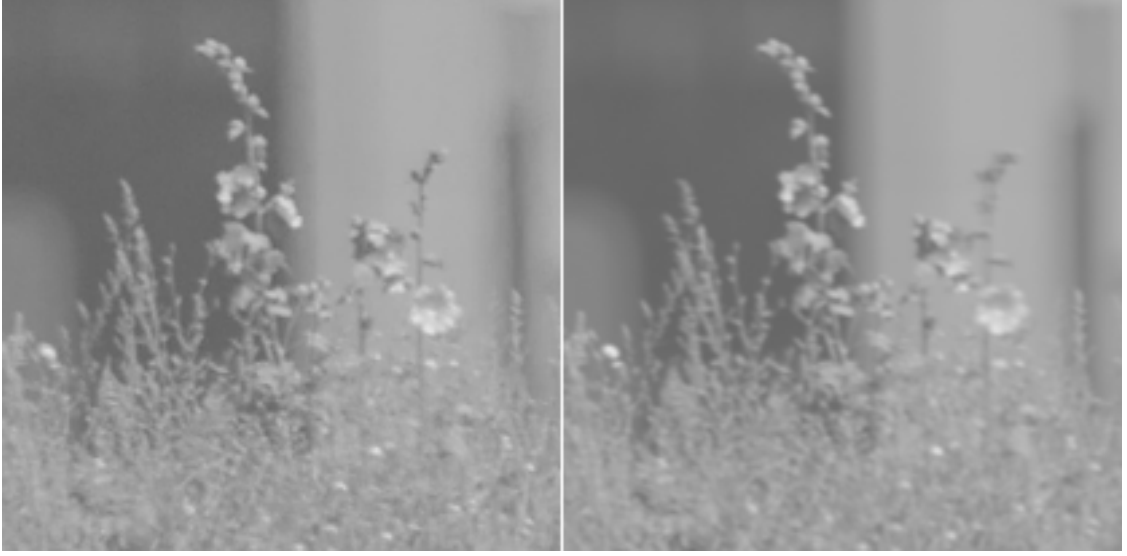
Şekil 3.5 Veri seti-4 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



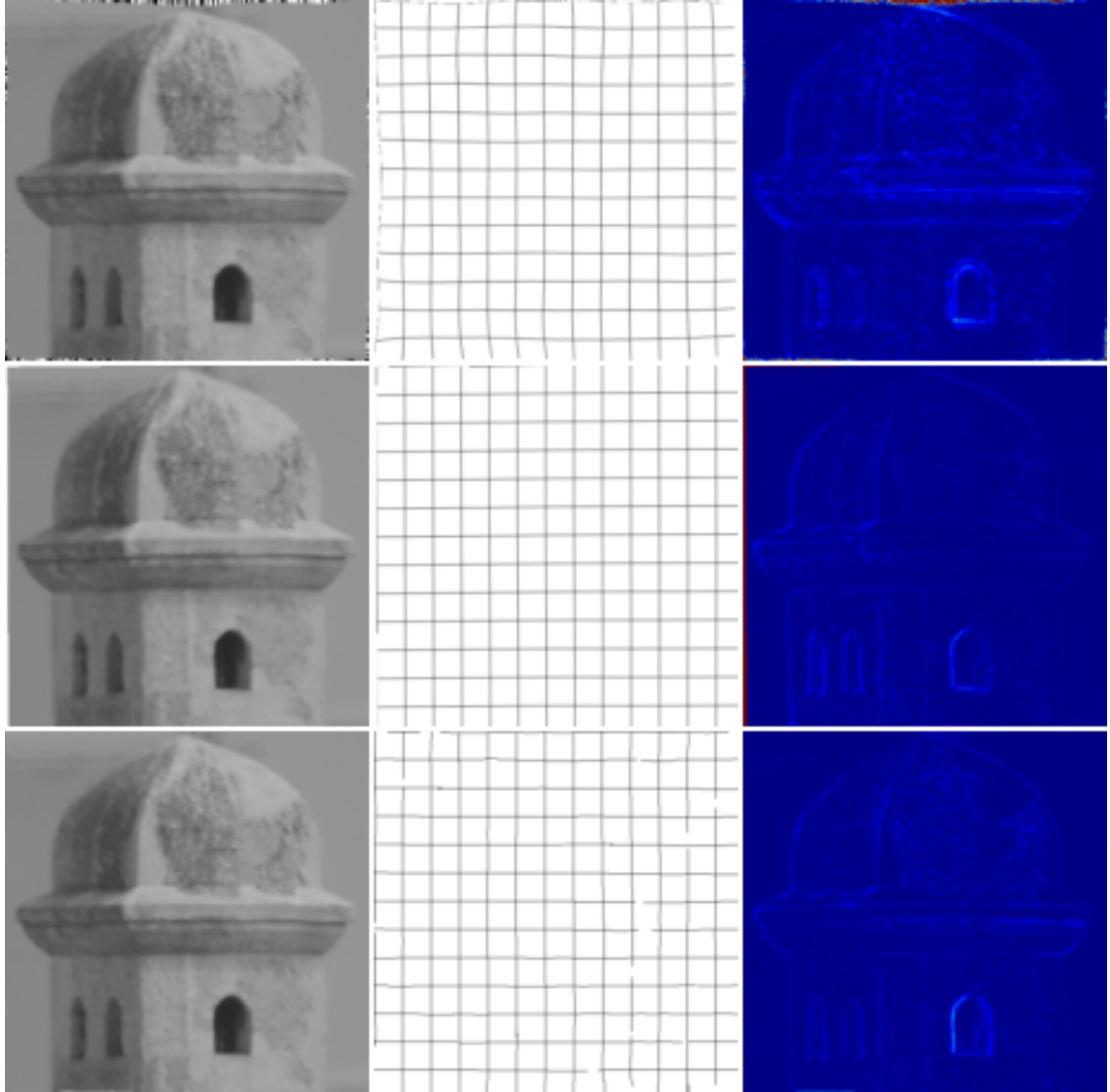
Şekil 3.6 Veri seti-7 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



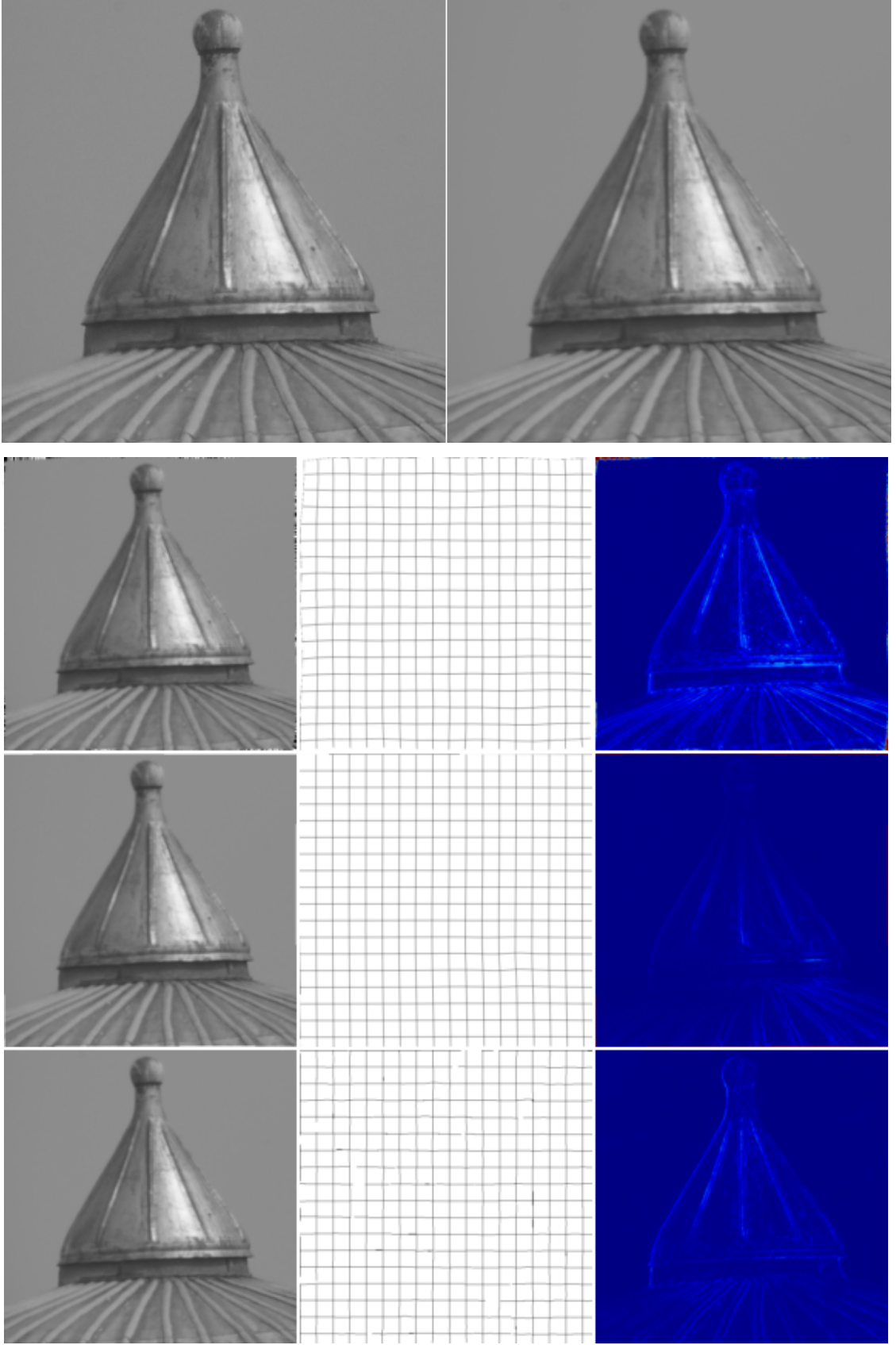
Şekil 3.7 Veri seti-8 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



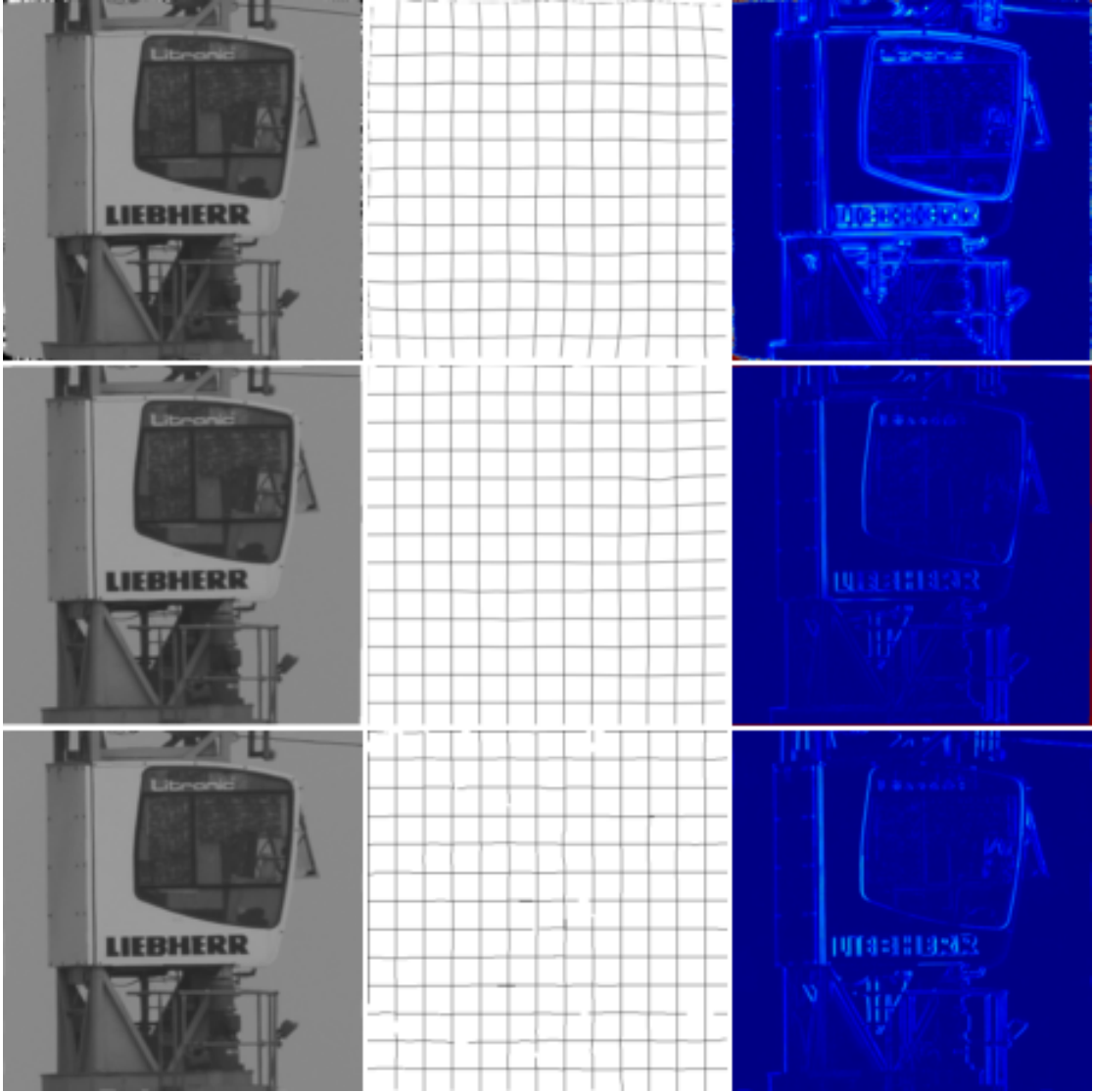
Şekil 3.8 Veri seti-9 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



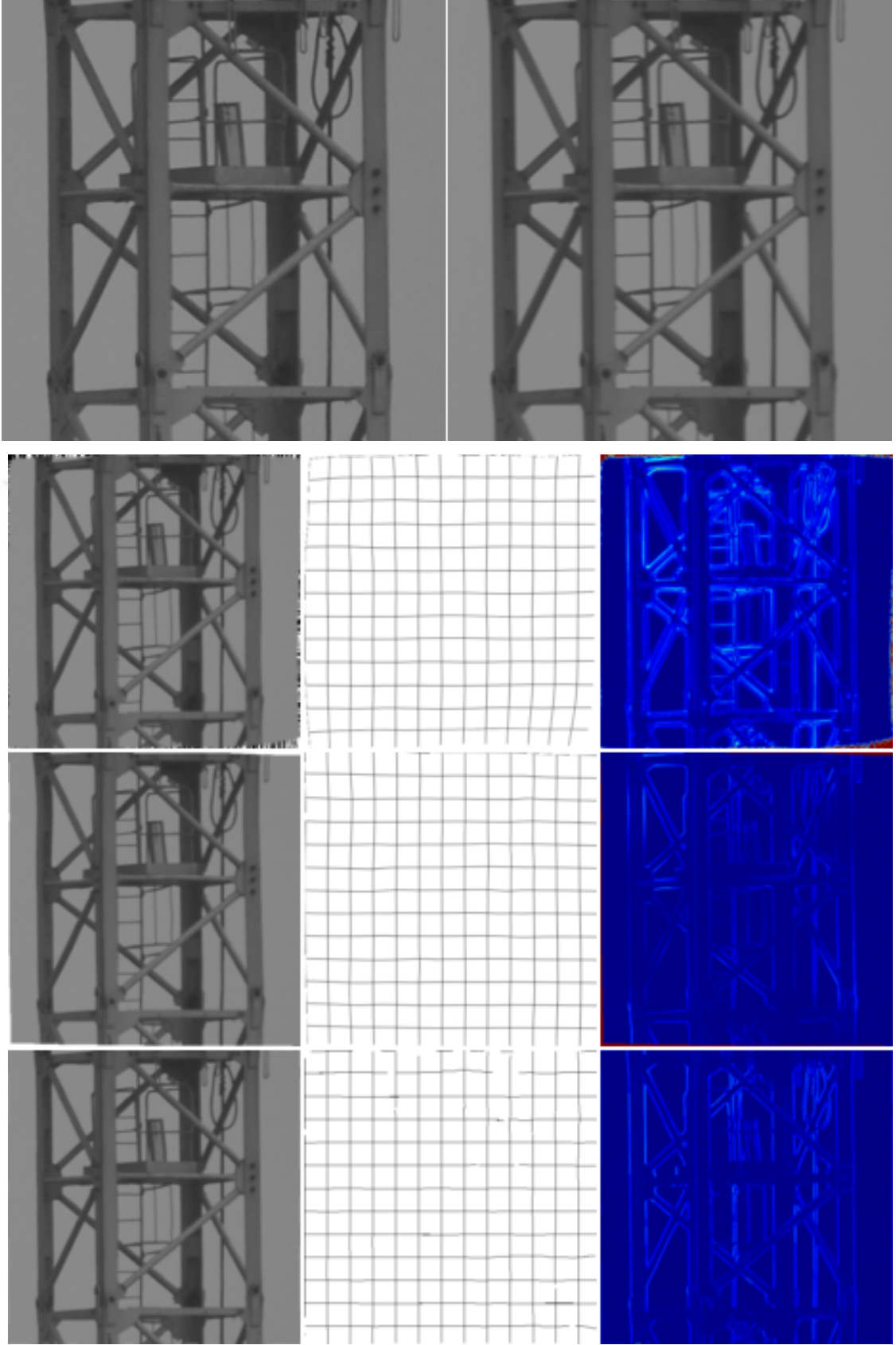
Şekil 3.9 Veri seti-13 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



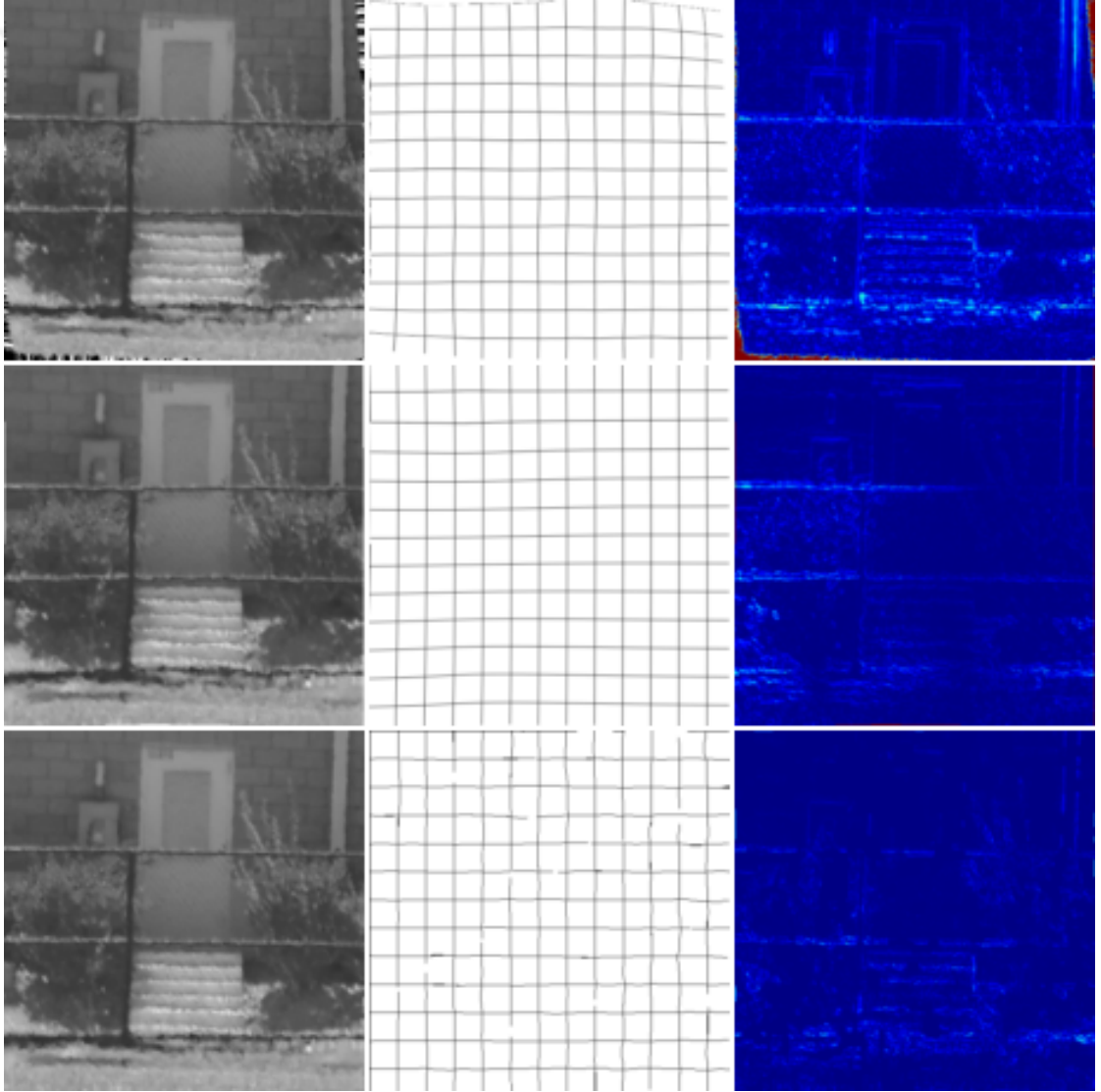
Şekil 3.10 Veri seti-14 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



Şekil 3.11 Veri seti-19 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi

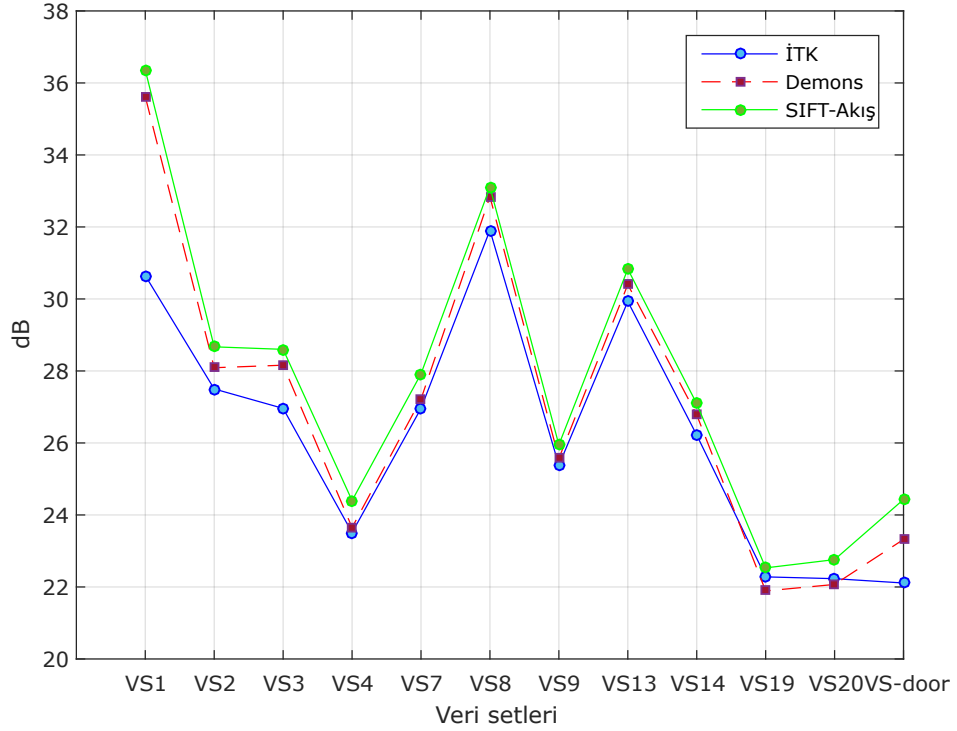


Şekil 3.12 Veri seti-20 için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi



Şekil 3.13 Door [3] için için çakıştırma sonuçları: 1.satır: (sol) Bozulmuş İmge, (sağ) Referans İmge, 2.-4. satırlar için: (sol) Çakıştırılmış İmge, (orta) Bozulma Haritası, (sağ) Bozulmuş ve Çakıştırılmış İmge Farkı, 2.satır: Demons Yöntemi, 3.satır: İTK Yöntemi, 4.satır: SIFT-Akış Yöntemi

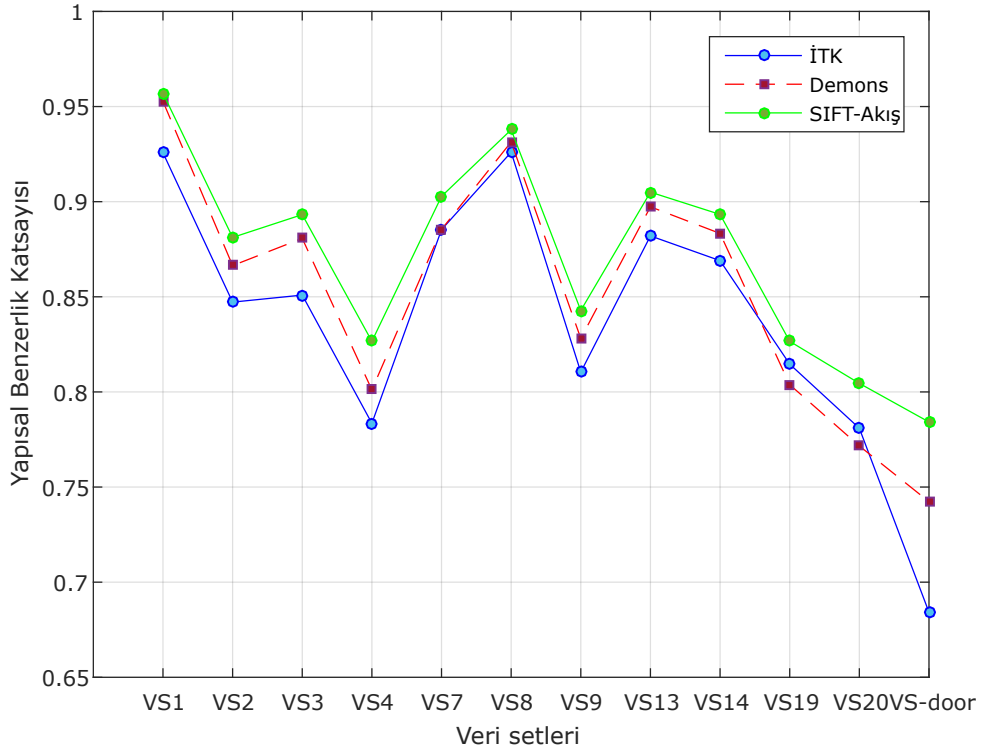
Şekil 3.14, Şekil 3.15, Şekil 3.16 ve Şekil 3.17’de sırasıyla PSNR, SSIM, NCC nesnel metrik sonuçları ve bu yöntemlerin işlem süreleri karşılaştırmaları grafik olarak sunulmuştur. İlk üç grafikten görüldüğü gibi en iyi sonuç SIFT-Akış yönteminden elde edilmektedir. Fakat işlem süresi grafiği SIFT-Akış yönteminin en yavaş yöntem olduğunu göstermektedir. Bu grafiğe göre en hızlı yöntem İTK yöntemidir.



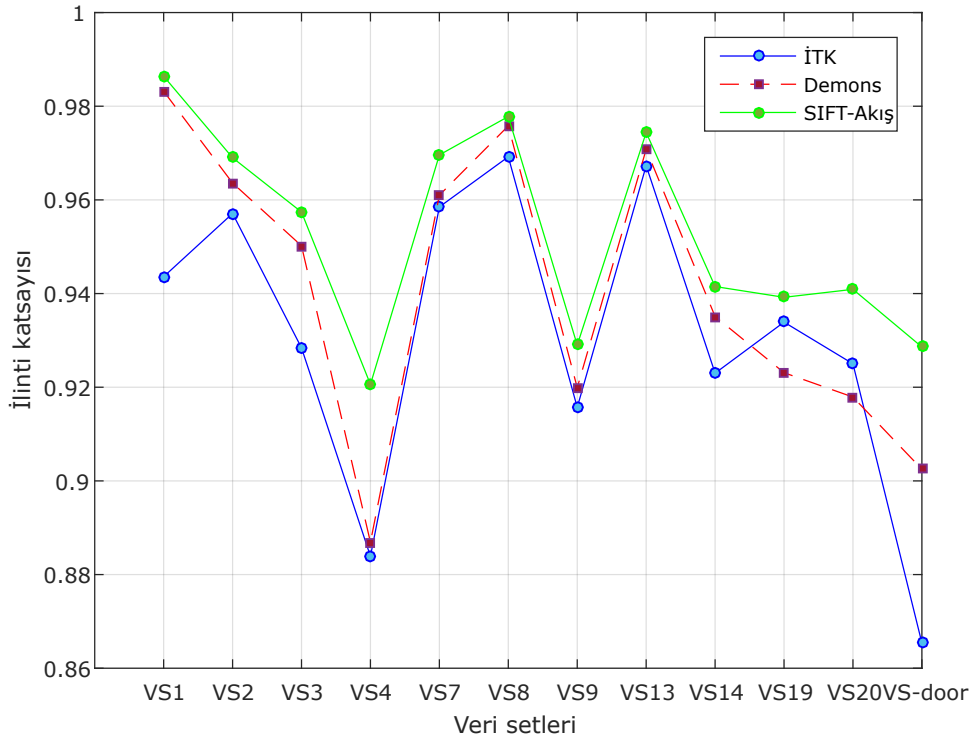
Şekil 3.14 Tepe sinyali gürültü oranı (PSNR) metrik sonuçları

Şekil 3.18’den Şekil 3.21’e kadar olan figürler belirli veri setleri üzerinde (Veri seti-1, Veri seti-3, Veri seti-4 ve Veri seti-8) hem işlenmemiş versiyonlarında hem de çakıştırılmış versiyonlarında bulunan bozulma bölgeleri ve bozulma şiddetleri görülmektedir. Şekillerden de görüleceği üzere çakıştırılmış versiyonlarında bulunan bozulma şiddetleri ve bozulma bölgelerinde gözle görülür şekilde bir azalma mevcuttur. Çakıştırılmış versiyonlar, işlenmemiş imgelerdeki geometrik deformasyonları önemli bir ölçüde azaltmıştır.

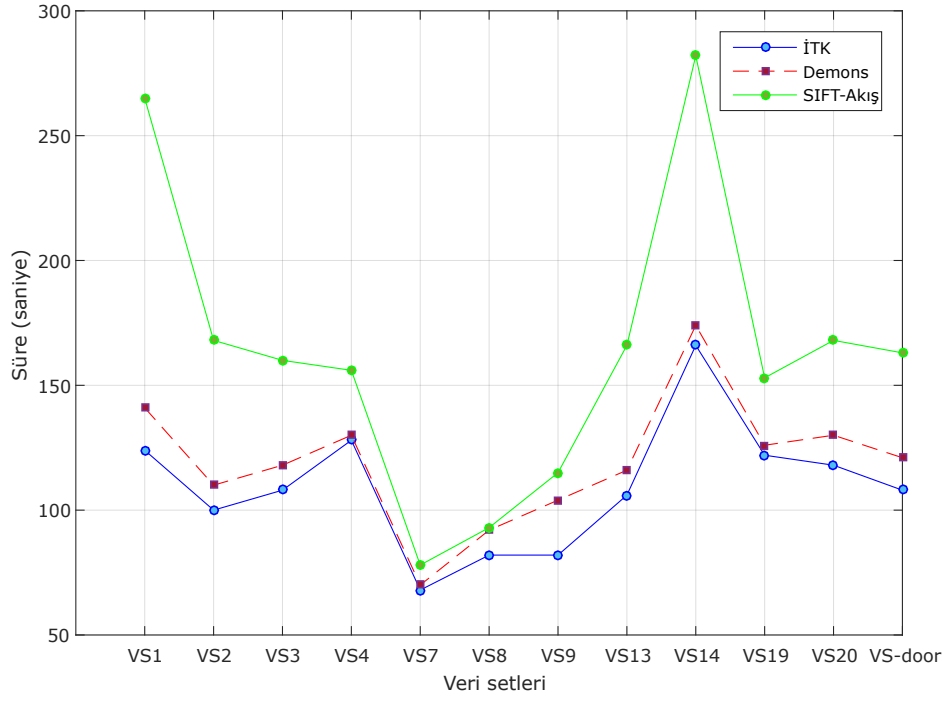
En iyi performansı SIFT-Akış yöntemi çıkardığı için imge kaynaştırma bölümünde SIFT-Akış yönteminden elde edilen çakıştırılmış imgeler kullanılmıştır.



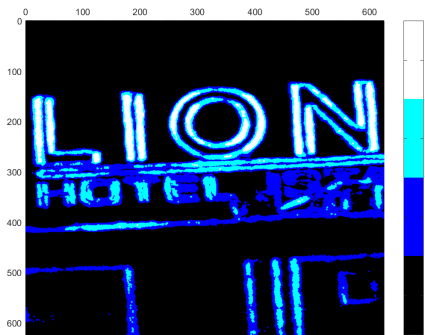
Şekil 3.15 Yapısal benzerlik indeksi ölçümü (SSIM) metrik sonuçları



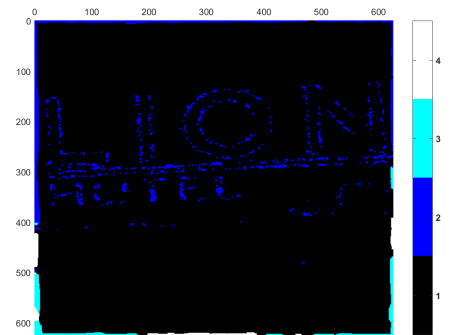
Şekil 3.16 İlinti katsayısı (NCC) metrik sonuçları



Şekil 3.17 Çakıştırma yöntemlerinin işlem süreleri

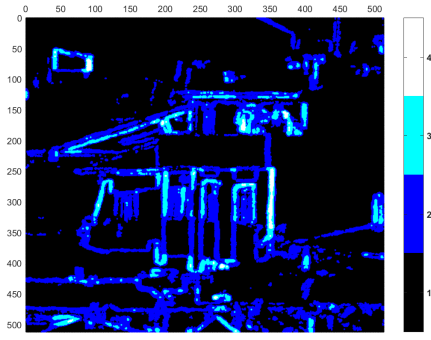


(a)

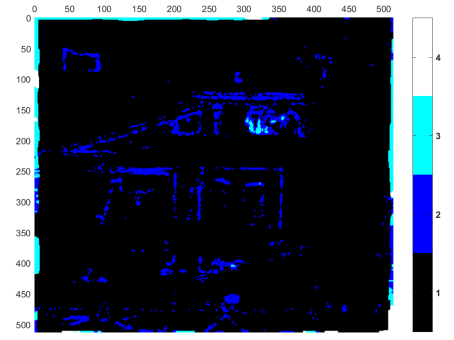


(b)

Şekil 3.18 Veri seti-1 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra

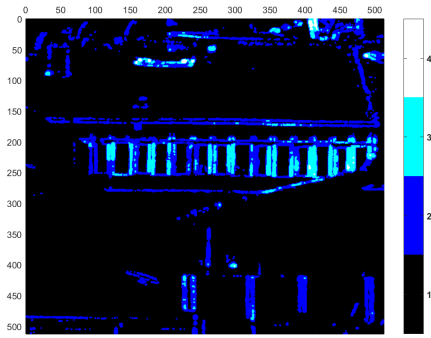


(a)

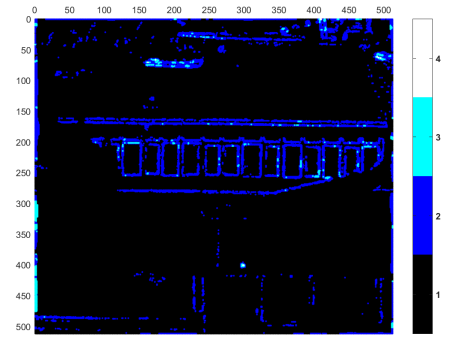


(b)

Şekil 3.19 Veri seti-3 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra

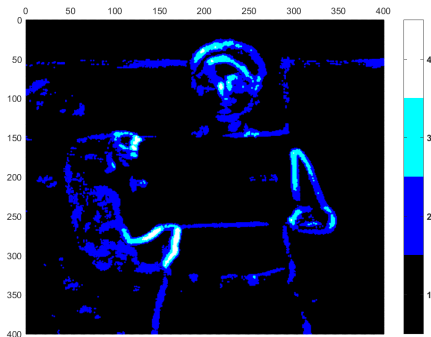


(a)

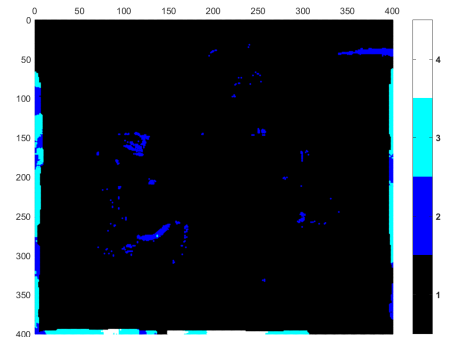


(b)

Şekil 3.20 Veri seti-4 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra



(a)



(b)

Şekil 3.21 Veri seti-8 için yerel dalgalanmaların görsel temsili a) Çakıştırma işleminden önce b) Çakıştırma işleminden sonra

4.1 Giriş

İmge kaynaştırma, birden fazla çakıştırılmış imge içerisinde piksel-seviyeli, özellik seviyeli ve karar seviyeli yöntemler kullanarak imge dizilerinden geometrik deformasyonu ve bulanıklığı azaltılmış tek bir imge elde edilmesini sağlar.

Veri kaynaştırmanın önemli alt alanlarından biri olan imge kaynaştırma, iki veya daha fazla imgeden ilgili bilgileri, giriş imgelerinden daha bilgilendirici ve ayrıntılı olacağı tek bir imgede birleştirme işlemidir. Bu işlem sonucunda elde edilen çıktı, görsel algı ve makine algılaması veya bilgisayar işleme için daha uygun olmalıdır. İmge kaynaştırmanın amacı, belirsizliği azaltmak ve çıktıdaki gereksiz, işe yaramayan bilgileri en aza indirmek ve aynı zamanda bir uygulama veya göreve özgü bilgileri üst düzeye çıkarmaktır [80].

Bu bölümde önerilen kaynaştırma işlemi olarak kullanılan makine öğrenmesi alt kolu olan evrişimli sinir ağları hakkında genel bilgiler verilmiş ve deneysel sonuçlar sunulmuştur.

Makine öğrenmesi, içinde derin öğrenme gibi birçok farklı yapay zeka alt kolu barındıran geniş bir uygulama alanıdır. Ortaya çıkma fikri "bilgisayarların, insan müdahalesi olmadan bir işlemi kendi başına öğrenebilmesi olası mıdır?", "Normal şartlarda girilen verilerden kurallar oluşturarak elde edilen anlamlı çıktılardan daha çok, kendi başına anlamlı çıktı oluşturabilmek için kuralları makinenin kendisi öğrenebilir mi?" gibi sorulardır. Kısaca makine, veri ve istenen çıktılar üzerinden o sistemin kurallarını öğrenir. Makine öğrenmesinde sistemler bir insan tarafından programlanmak yerine eğitilirler (training). Verilen örnekler esas alınarak matematiksel işlemlerle sonuca yönelik kurallar otomatik olarak oluşturulur.

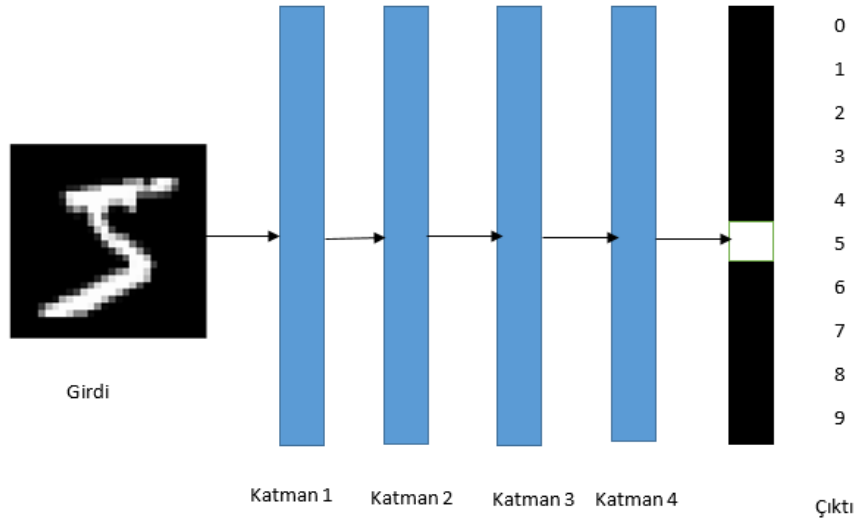
Makine öğrenmesi için şu parametreler olmalıdır:

- **Veri Girişi:**Belli bir sistemi çalıştıracak kuralları oluşturabilmek için veri setinin

olması gereklidir.

- **İstenen Çıktı:**Örneğin imge tanıma probleminde imgenin doğru sınıflandırılması istenen çıktı olarak kabul edilir.
- **Başarı Ölçümü:**İstenen çıktı ile algoritmanın bulduğu çıktı arasındaki farkın ölçülmesi gerekmektedir. Algoritma bu farkı minimize etmek için geri besleme olarak bu ölçümü kullanır. "Öğrenme" kavramı da tam olarak budur.

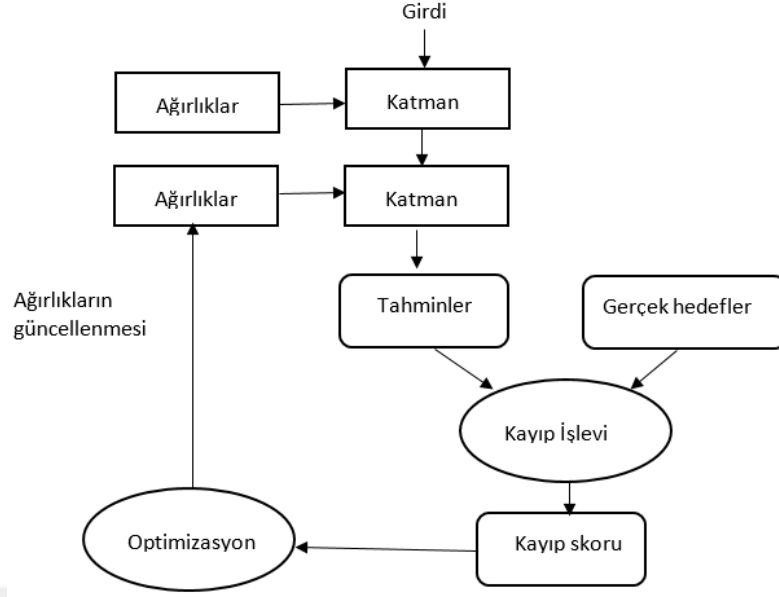
Derin öğrenme, peş peşe sıralanmış katmanlarda veriler işlenirken her bir katmanda daha yararlı özellikler ortaya çıkaran makine öğrenmesi çeşididir. Derinliğin ölçütü, sistemdeki katman sayısıdır. Şekil 4.1'de; giriş verisinden (örneğin imge), istenen sonuca örneğin gerçek etiket değerine eşleşebilen bir derin ağ örneği görülmektedir. Bu sonuca ulaşabilmesi için sistemde bir çok veri dizisinin işlenmesi gerekmektedir.



Şekil 4.1 Derin ağ örneği

Derin ağın çıktısını kontrol edebilmek için istenen çıktı ile algoritma çıktısı arasındaki farkı ölçmek gerekir. Bu işlem ağın kayıp işlevi tarafından yapılır. Kayıp işlevi, ölçümü yapar ve başarı puanları. Ardından bu başarı puanı yani kayıp skorunu geribesleme olarak kullanıp mevcut veri için amacı gerçekleştirmek adına (kayıbı azaltacak şekilde) ağırlık matrisinin değerini günceller. Bu işleme geriye yayılım (back propagation) denir ve bu da optimizasyon algoritmalarının görevidir. Şekil 4.2'de bu parçalar arasındaki ilişki görülmektedir.

Ağırlık matrisine başlangıç olarak rastgele değerler verilir. Bu nedenle ağ, etkili dönüşümler yapamaz. Sonuçta çıktı, istenen değerden çok farklı olacağından



Şekil 4.2 Ağ, katmanlar, kayıp işlevi ve optimizasyon arasındaki ilişki

kayıp skoru yüksek çıkacaktır. Ağ, her adımda veriyi işlediğinde ağırlıklar yeniden değerlendirilecek ve bu şekilde kayıp skoru azalacaktır. Öğrenme döngüsü olarak bilinen bu işlem her adımda tekrarlandığında kayıp işlevi en düşük değerine ulaşacaktır [81].

Derin öğrenmenin bir çok alanda popüler olmaya başlamasının nedeni bir çok problem için verimli sonuçlar üretmesidir. Bu sonuçları oluşturabilmesinin nedeni veriler işlenirken öznetelik çıkarımı denilen adımı otomatik olarak yapmasıdır. Daha önceki algoritmalarda insanlar verileri işlerken, uzun süren önışlemler (daha iyi modelleme yapabilmek için) yapmak zorundalardı. Öznetelik çıkarma denilen bu işlemde ise derin öğrenme, öznetelikleri kendiliğinden öğrenerek otomatik olarak yapar.

Derin öğrenmenin son 10 yılda daha çok uygulama alanında kullanılmasının nedenleri şu şekilde sıralanabilir:

- 1) Donanım
- 2) Veri Setleri
- 3) İyileştirilmiş Algoritmalar

1) **Donanım:** Milenyumdan itibaren ekran kartı üreten firmalar (NVIDIA ve AMD) bilgisayarların yüksek çözünürlüklü oyunlar ve karmaşık hesaplamalar için ihtiyaç duyduğu paralel çalışan ekran kartı çekirdeklerinin (GPU) geliştirilmesi için önemli derecede yatırım yapmışlardır. NVIDIA firmasının CUDA adında kendi ekran kartı üzerinde programlama yapılabilmesini sağlayan arayüzün geliştirilmesiyle bilim insanları, mühendisler, üniversite öğrencileri ve amatör ilgililer de bu yatırımdan

yararlanmaya başlamışlardır. Son 10 yıldır Ciresan [82] ve Krizhevsky'nin [83] öncülük ettiği birçok araştırmacı, sinir ağlarını CUDA ile eğitmeye başlamışlardır.

2) **Veri Setleri:** Son 20 yılda internetin yaygınlaşmasıyla derin öğrenmenin ihtiyacı olan büyük veri setlerinin oluşturulması, toplanması ve paylaşılmasının yanında bu büyük verilerin depolanmasını sağlayan ürünlerin gelişmesiyle, veri seti derin ağlar için yeterli düzeye gelmiştir. Örneğin video işlemede Youtube; imge işlemede Facebook, Instagram; doğal dil işleme alanında Wikipedia büyük bir veri kaynağıdır.

3) **Algoritmalar:** Son 10 yıl içerisinde araştırmacılar tarafından oluşturulan çok önemli birkaç algoritmanın bilim çevresinde paylaşılmasıyla kayıp skorunun geriye daha etkili yayınımları sağlanmıştır. Bunlar:

- Derin ağ katmanlarında kullanılacak daha verimli aktivasyon fonksiyonları
- Daha uygun ilk ağırlıklandırma yöntemleri
- Daha uygun optimizasyon yöntemleri

olarak sıralandırılabilir.

Böylece yığın normalizasyonu [84], artık bağlantılar [85], derinlemesine evrişim [86] gibi hatanın geriye daha iyi yayınımları sağlayan yöntemler ortaya çıkmıştır.

Derin öğrenmenin ana bileşeni, girdi için filtre olarak kullanılan katmanlardır. Veri, katmana işlenmemiş olarak girer ve daha uygun bir şekilde çıkar. Öğrenme işlemi genelde, basit katmanları üst üste getirerek verinin ilerledikçe daha da saflaştırılmasını sağlayan bir süreçtir.

4.2 Makine Öğrenmesi Sınıfları

Makine öğrenmesi uygulamaları genelde 4 sınıfa ayrılır.

4.2.1 Denetimli Öğrenme

En çok kullanılan sınıftır. Veri setindeki giriş bilgilerinden hedeflere olan eşleştirmeyi öğrenir. İmge sınıflandırma, ses tanıma gibi uygulamalar bu sınıfa dahildir. Genelde denetimli öğrenme, sınıflandırma ve bağlanım içerse de bu sınıfın başka uygulama alanları da mevcuttur. Örneğin:

Dizi Oluşturma: Bu uygulama genelde arka arkaya sınıflandırma olarak bilinir. Örnek uygulama: mesaj yazarken arka arkaya gelen kelimeleri kestirmek bu alan için doğru

bir örnektir.

Nesne Tanıma: Verilen imgede, nesnelere tespit etmek bu alanın çalışma konusudur. Burada hem sınıflandırma, hem de çerçeve içine alınan nesnenin koordinatlarının vektör bağlantısını ile kestirimini içeren hibrid bir yapı bulunur.

İmge Bölütleme: Bu alanda amaç, imgede nesneye özel piksel seviyesinde maske oluşturmaktır. Örneğin akciğer bilgisayarlı tomografi imgelerinin bölütleme uygulaması gibi bu uygulama alanına verilebilecek uygun bir örnektir.

4.2.2 Denetimsiz Öğrenme

Bu sınıfın hedefi verileri sıkıştırmak, verilerdeki gereksiz bilgileri azaltmak için girdilerden herhangi bir bilinen hedefin aracılığı olmadan uygun dönüşümler bulmaktır. Denetimsiz öğrenme veriyi anlamak için kullanılan bir aşamadır. Boyut indirgeme, kümeleme gibi uygulamalarda kullanılır.

4.2.3 Yarı Denetimli Öğrenme

Denetimli öğrenmenin özel bir durumudur. Tek farkı; denetimli öğrenmede öğrenmeyi birinin denetlemesi gerekir. Yarı denetimlide insan faktörü yoktur. Hâlâ denetim vardır fakat bu, girdilerden sezgisel olarak oluşturulur [81].

Otokodlayıcılar bu bölümdeki en çok bilinen uygulamadır. Hedefler girdilerden öğrenilir. Yarı denetimli öğrenme aslında uygulanış şekline göre bazen denetimli, bazen de denetimsiz grubuna dâhil edilebilir.

4.2.4 Pekiştirmeli Öğrenme

Çok özel bir uygulama alanına sahiptir. Sadece bazı oyun uygulamalarında kullanılmıştır.

4.3 Öğrenme Modellerinin Değerlendirilmesi

Bir sinir ağı oluştururken veri seti 3 kısma ayrılır. Bunlar eğitim, doğrulama ve test veri setleridir. Veri setini üçe ayırmanın nedeni geliştirilen modelin aşırı öğrenme olayına girmesini engellemektir. Eğitim boyunca her epokta sürekli artan performans, daha önce karşılaşılmayan verilerde performans düşüşüne neden olur. Burada asıl amaç, modellerin genelleştirilmesidir. Yani modelin bütün veri seti çeşitleri üzerinde başarılı olmaları istenir. Fakat önünde bir engel vardır. Bu da aşırı öğrenmedir.

4.3.1 Eğitim, Doğrulama ve Test Veri Setleri

Model değerlendirmesi bu 3 set üzerinden yapılır. Eğitim veri setinde model eğitilir ve doğrulama setinde değerlendirilir. Son aşamada ise test verisi üzerinde modelin başarımı denenir.

Modeli geliştirirken modelin yapılandırılmasında hassas ayar yapılır. Örneğin katman sayısı ve büyüklüğünün (modelin hiperparametreleri) değiştirilmesi gibi ayarlar bu süreçte yapılır. Bu hassas ayar, modelin doğrulama aşamasında verdiği geri bildirimle göre ayarlanır. Bu işlem de bir nevi öğrenmedir. Parametre uzayında optimal parametre değerleri aranır. Böylece; model doğrulama seti üzerindeki eğitilmemiş veriler üzerinden performansına bakılarak değerlendirilir.

Doğrulama seti üzerindeki performansa göre yapılan her hassas ayarda doğrulama setinden modele doğru bir bilgi sızıntısı meydana gelir. Bu ayar, bir parametre için bir kere yapılırsa çok az bir sızıntı modele geçeceğinden modelin hâlâ güvenilir olduğu söylenebilir. Fakat her denemeden sonra doğrulama setinde değerlendirme yapıp sonuca göre model güncellenirse doğrulamadan modele doğru giderek artan bir sızıntı oluşur ve hem eğitim hem de doğrulama veri setinde gerçek olmayan, performansı şişirilmiş bir model oluşur.

Aslında değerlendirmenin test verisi gibi daha önce hiç bilinmeyen bir set üzerinde yapılması gerekir. Modelin test veri seti performansına göre güncellenmesi genelleştirmeyi olumsuz yönde etkiler. Eğer veri seti az ise şu şekillerde değerlendirme yapılabilir.

4.3.1.1 Hold-Out Doğrulama

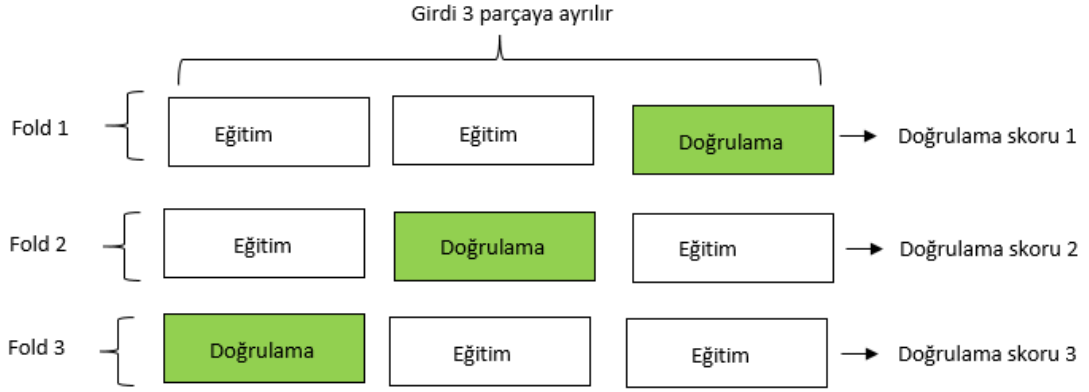
Veriler eğitim ve test olarak ayrılır. Eğitim seti eğitilir ve model test veri seti üzerinde doğrulanır. Sızıntıyı engellemek için test veri setinde model güncellenmemelidir.

Bu doğrulama yönteminin bir dezavantajı vardır. O da, veri setinin az olduğu sınır ağlarında istatistiksel olarak veri setinin gösterimini sağlayamayan doğrulama ve test veri seti oluşur. Bununla başa çıkabilmek için K-Fold doğrulama ve döngüsel K-Fold doğrulama yöntemleri geliştirilmiştir.

4.3.1.2 K-Fold Doğrulama

Veriler k eşit parçaya bölünür. Her bir parçada kalan $k - 1$ için eğitim yapıp o parçada değerlendirilir. En sonunda bütün doğrulama skorlarının ortalaması alınır. Bu yöntemle değerlendirme veri setiyle hassas ayar yapılabilir. Şekil 4.3'te sistemin

blok diyagramı görülmektedir.



Şekil 4.3 K-fold doğrulama

4.3.1.3 Döngüsel K-Fold Doğrulama

Bu yöntem veri seti az ve modeli en yüksek hassasiyetle değerlendirmek için kullanılır. K-fold doğrulamanın birden fazla kullanımı esnasında her defasında karıştırılır.

4.3.2 Veri Önışleme ve Öznitelik Çıkarma

4.3.2.1 Sinir Ağları için Veri Önışleme

Ham verileri katmanlara göndermeden önce; vektör haline getirme, normalizasyon işlemi uygulama ve niteliklerin vektör halini alması gibi sinir ağlarının kullanabileceği hâle getirmek gerekir.

Verinin Vektöre Dönüştürülmesi

Sinir ağlarında bulunan tüm işlenmemiş veriler ve hedefler ondalık sayı tensörlerinde olmalıdır. Ses, imge, video veya metin olsun; bütün veri setleri için ilk aşama veriyi vektör haline getirmektir. Genelde yapılan işlem veriyi float32 veri tipinde tensörlere çevirmektir. Eğer veri zaten vektör biçimindeyse bu işlem yapılmaz.

Normalizasyon

Veri eğer imgeyse, sinir ağını beslemeden önce piksellerin 0-255 arasında gri değerleriyle başlanır. Başlamadan önce tüm piksel değerleri 255'e bölünüp float32 veri tipinde 0-1 aralığında ondalık sayılara dönüştürülür.

4.3.2.2 Öznitelik Çıkarma

Öznitelik çıkarımı, mevcut veri hakkında toplanan bilgilerin sistemin daha iyi sonuç vermesi için makine tarafından henüz öğrenilmemiş yararlı bilgilerin manuel olarak modele verilmesidir. Eldeki verilerin algoritmanın öğrenmesine katkı sağlayacak şekilde modeli beslemesi gerekir.

4.3.3 Aşırı Öğrenme ve Eksik Öğrenme

Makine öğrenmesinde model, eğitim verilerinden elde edilen bilgilerle oluşturulur. Bu işlem sonucunda iki sonuç ortaya çıkabilir. Model ya aşırı öğrenmiştir ya da eksik öğrenmiştir. Bu sonuç modelin genelleştirilememesine ve bu nedenle de yaptığı kestirimlerde hata oranının büyük çıkmasına neden olacaktır. Bütün makina öğrenmesi modelleri aşırı öğrenmeye meyillidir. Önemli olan aşırı öğrenmeyi engellemektir.

Makine öğrenmesinde modellerde olması istenen iki esas husus, eniyileme ve genelleştirmedir. 'Eniyileme'; eğitim veri setinde performansı yükseltmek için doğrulama veri seti aracılığıyla modeli güncellerken (öğrenme süreci) 'genelleştirme' ise modelin, hiç işlemediği veri setleri üzerinde de yüksek performans göstermesidir. Burada asıl husus, genelleştirmeyi iyileştirmek'tir.

Modelin anlamsız ve sonucu yanlış yönlendirecek bilgileri öğrenmesi istenmeyen bir durumdur. Bunu aşmak için, model daha çok veri setiyle beslenmelidir. Eğitim için daha çok veri setinin olması genelleştirmenin daha iyi olacağı anlamına gelir. Eğer eğitim veri setine müdahale edilemiyorsa o halde modelin öğreneceği bilgi miktarına bakılır. Bu da modelin katmanlarda daha önemli öznitelikleri öğrenmesini sağlayıp genelleştirme performansını iyileştirecektir. Aşırı öğrenmeyi engellemek için sisteme düzenleme eklenmelidir.

4.3.3.1 Sınır Ağını Küçültmek

Bir sistemde aşırı öğrenme istenmeyen bir durumdur. Bunu engellemenin en kolay yolu modelin derinliğini azaltmaktır. Katman sayısı arttıkça derinlik ve buna bağlı parametre sayısı da artmaktadır. Bu da modelin sistemi ezberlemesine neden olur ve eğitim veri setinde oldukça yüksek bir performans gösterir. Fakat istenilen durum sadece eğitim veri setindeki performansının iyi olması değil, aynı zamanda hiç görmediği veri setinde de başarılı olmasıdır. Yani genelleştirilebilir olmasıdır.

Diğer taraftan modelin öğrenmesi için eldeki veri seti yeterli değilse, giriş-çıkış arasındaki bağlantıyı öğrenemeyecek, bu nedenle model de genelleştirilemeyecektir. Burada parametre sayısının modelin aşırı öğrenmesine neden olacak şekilde çok

olmaması gereklidir. Modelin parametre sayısını belirleyen katman sayısı ve birim büyüklüğü ile ilişkili bir formül bulunmamaktadır. Bundan dolayı eğitim setinde eğitilen model doğrulama veri setiyle hassas ayarı yapılarak en uygun büyüklüğün seçilmesi gerekmektedir. Böylece, küçük katmanlar ve az parametre ile modele başlanıp doğrulama veri setiyle kaybın azaldığı görülerek yeni katmanlar eklenmelidir.

4.3.3.2 Ağırlıkların Düzenleştirilmesi

Aşırı öğrenmeyi engellemenin bir diğer yolu da derin ağlara giden sınırlara atanan ağırlıkların değerlerini küçültmek, dağılımı düzenli hale getirmektir. Kayıp işlevine yeni bir maliyet getirerek ağırlıkların büyük değerler alması önlenir.

L-1 Normu: Maliyet, ağırlıkların katsayılarının mutlak değerlerine eklenir.

L-2 Normu: Maliyet, ağırlıkların katsayılarının karelerine oransal olarak eklenir. Bazı uygulamalar için de L-1 ve L-2 birlikte kullanılabilir [81].

4.3.3.3 İletim Sönümü

İletim sönümü (dropout) derin ağların düzenlenmesinde en etkili yöntemdir. Hinton vd. [87] tarafından evrişimli sinir ağları üzerinde uygulanması önerilmiştir. İletim sönümü eklenen katmanda bilgilerin bir kısmı sıfırlanır. Genellikle bilgilerin 0.2 veya 0.5 aralığında bir oranı seçilir ve sıfırlanır. Test verilerine bu teknik uygulanmaz. Fakat test zamanı katmanın çıktısı iletim sönüm oranı kadar küçültülür.

4.4 Evrişimli Sinir Ağları

Makine öğrenmesi alanında imge işleme uygulamalarında en çok kullanılan yöntem evrişimli sinir ağlarıdır. Şekil 4.4'te örnek bir evrişimli sinir ağı modeli ve Şekil 4.5'de ise bu ağ modelinin içerdiği yapılar görülmektedir. Evrişimli sinir ağları iki şekilden de görüleceği gibi evrişim (Conv2D), en büyükleri biriktirme (MaxPooling2D) ve tamamen bağlı katman (Dense)'dan oluşur.

4.4.1 Evrişim İşlemi

Tamamen bağlı katmanların aksine evrişimli katmanlar yerel örüntüleri öğrenmektedir. İmgeler için örüntüden kasıt filtre boyutunun izin verdiği ölçüde öznitelik çıkarmadır. Eğer filtre boyutu 5x5 ise imgede bu ölçülerde bilgiler öğrenilecektir.

```

model = models.Sequential()
model.add(layers.Conv2D(16, (3, 3), activation='relu', input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(32, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

```

Şekil 4.4 Örnek bir evrişimli sinir ağı modeli

| Layer (type) | Output Shape | Param # |
|--------------------------------|----------------------|---------|
| conv2d_4 (Conv2D) | (None, 148, 148, 16) | 448 |
| max_pooling2d_3 (MaxPooling2D) | (None, 74, 74, 16) | 0 |
| conv2d_5 (Conv2D) | (None, 72, 72, 32) | 4640 |
| max_pooling2d_4 (MaxPooling2D) | (None, 36, 36, 32) | 0 |
| conv2d_6 (Conv2D) | (None, 34, 34, 64) | 18496 |
| flatten_1 (Flatten) | (None, 73984) | 0 |
| dense_1 (Dense) | (None, 64) | 4735040 |
| dense_2 (Dense) | (None, 10) | 650 |
| Total params: 4,759,274 | | |
| Trainable params: 4,759,274 | | |
| Non-trainable params: 0 | | |

Şekil 4.5 Örnek sinir ağı modelinin yapısı

Evrişimli sinir ağlarının imge işleme alanında performansının yüksek olmasının nedeni evrişim işleminde öğrenilen bir örüntü imgenin her tarafında anlaşılabilir ve tanınabilir. Bu da daha az veri seti ile performansı yüksek çıktıları ortaya çıkarır.

Tamamen bağlı katmanda öğrenme küresel olduğundan bir bölgede öğrendiği bilgiye farklı bir bölgede tekrar karşılaşırsa bunun için de yeniden hesaplama (öğrenme) yapması gerekecektir. Bu da veri setinin fazla olmasını gerektirir.

Evrişimli sinir ağlarında ilk katmanda öğrenilen örüntüler genelde kenar ve köşe gibi küçük bölgelerdir. İkinci katman ise ilk katmandan öğrenilenler üzerinden yeniden evrişim işlemiyle daha geniş nitelikler çıkarır. Ağ derinleştikçe özneliklerin çıkarımı zorlaşır ve soyut görseller öğrenilmeye başlanır.

Evrişim işlemi veri olarak 3 boyutlu bir tensörle çalışır. Buna nitelik haritası denir. Tensörde (genişlik, yükseklik, kanal sayısı) bilgileri bulunmaktadır. Eğer imge renkli (RGB-24 bit) ise kanal sayısı 3, gri seviyeli ise kanal sayısı 1'dir. Evrişim, veri setinden evrişim boyutu kadar (filtre boyutu) bölgesel parçalar alır ve dönüşümle nitelik haritası oluşturur.

Örneğin yukarıdaki örnek sinir ağında giriş verisi (150,150,3) boyutundadır ve 16 adet (3x3) boyutunda Conv2D filtreyle (148,148,16) boyutunda bir nitelik haritası çıkarır. Burada çıkış nitelik haritası [(imge yüksekliği-filtre yüksekliği+1), (imge genişliği-filtre genişliği+1), filtre sayısı] yani (150-3+1),(150-3+1),16=(148, 148,16) şeklinde hesaplanır.

Evrişimde önemli olan 2 husus vardır.

Veriden çıkarılacak örüntülerin boyutu: (3x3), (5x5) veya (7x7) olabilir. Nitelik haritasının derinliği; Evrişimde kaç tane filtre kullanılacağını gösterir. Bu nedenle Conv2D içerisine bu iki parametre mutlaka belirtilmelidir.

Evrişim katmanında yapılan işlem bütün veriden 3x3 veya 5x5 boyutunda pencereleri birer birer kaydırarak 3 boyutlu parçalar alır ve sonra tüm bu 3 boyutlu parçalar 1 boyutlu vektöre dönüştürülür. Toplam filtre sayısı ile birlikte ortaya ilk katmanın nitelik haritası çıkar. Burada giriş verisi ile çıktı haritasının boyutları aynı olmayabilir. Bunun 2 nedeni vardır.

1) Kenar Etkisi

2) Adım Aralığı

4.4.2 Kenar Etkisi ve Doldurma

5x5'lik bir giriş verisi üzerinde filtre boyutu 3x3 olan toplam 9 nitelik çıkarımı yapılabilir $((5-3+1) \times (5-3+1)) = 3 \times 3 = 9$. Eğer girdi verisinin ve çıktı nitelik haritasının boyutunun aynı olmasını istiyorsak "doldurma" adlı işlemi uygulamalıyız. Bu işlem girdinin her yanına istenen miktarda sıfırlardan oluşan satır ve sütunlar ekleyerek tüm verinin evrişim işleminden geçmesini sağlar. 3x3'lük bir pencere için 1 satır ve 1 sütun, 5x5'lik bir bir filtre için 2 satır ve 2 sütun eklenmelidir. Conv2D katmanlarında doldurma "padding" ile tanımlanır. "valid" veya "same" ile istenilen durum belirtilir.

4.4.3 Evrişim Adım Aralığı

Çıktının girdi ile aynı boyutta olmamasının bir diğer nedeni adım aralığıdır. Filtre veriyi 3 boyutlu tensörlere parçalarken birer birer kayar. Yani iki evrişim işlemi arasındaki adım aralığı 1'dir. Adım aralığı 1'den büyük olabilir. Adım aralığının 2'ye çıkarılması çıktının boyutunun en az yarıya düşmesine (adım aralığından dolayı ve kenar etkisi) neden olur. Genelde büyük girdilerde hesaplama maliyetini düşürmek için boyut küçültmek yerine en büyükleri biriktirme işlemi yapılır.

4.4.4 En Büyükleri Biriktirme

Sinir ağında her MaxPooling2D işleminde çıktı boyutu yarıya iner. Bu kısmın görevi aşırı öğrenmeyi engelleyip nitelik haritasının boyutunu küçültmektir.

Bu işlemde girdi nitelik haritasından pencereler çıkarıp her kanalın en büyük değerini alır. Yani maks. kuralıyla çalışır. Bu süreçte çıktı girdinin yarı boyutuna indirgenmek isteniyorsa (2x2)'lik pencerelerle, girdinin 4'te 1'i olarak isteniyorsa (4x4) boyutunda pencerelerle en büyükler biriktirilir. Çıktı haritalarının küçültülmesinin 2 nedeni vardır.

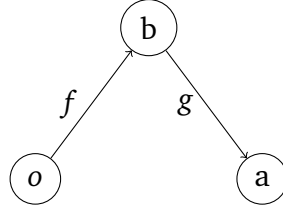
1) Ard arda gelen Conv2D evrişim katmanları her bir katmanda parametre sayısını daha da çok arttıracaktır. MaxPooling2D işlemi olmadan evrişim işlemi ve ardından tamamen bağlı bir katman eklenirse (örneğin 512 boyutunda Dense) bu sayı milyarlarca parametre anlamına gelecektir. Küçük bir model için bu sayı çok fazla ve ağ aşırı öğrenme yoluna gidecektir.

2) Uzamsal hiyerarşileri öğrenemez [81]. Esasında en büyükleri biriktirip çıkış boyutunu küçültmenin amacı daha az parametreyle işlem yapmak (maliyeti düşürmek, aşırı öğrenmeyi engellemek) ve evrişim katmanındaki filtrelerin her defasında daha büyük pencerelere bakmasını sağlamaktır.

Çıktı haritasının boyutu adım aralığı yoluyla da küçültülebilir. Fakat ağ için en etkili yöntem katmana MaxPooling2D işleminin yapılmasıdır. Böylece daha önemli bilgiler elde edilir.

4.5 Otokodlayıcılar

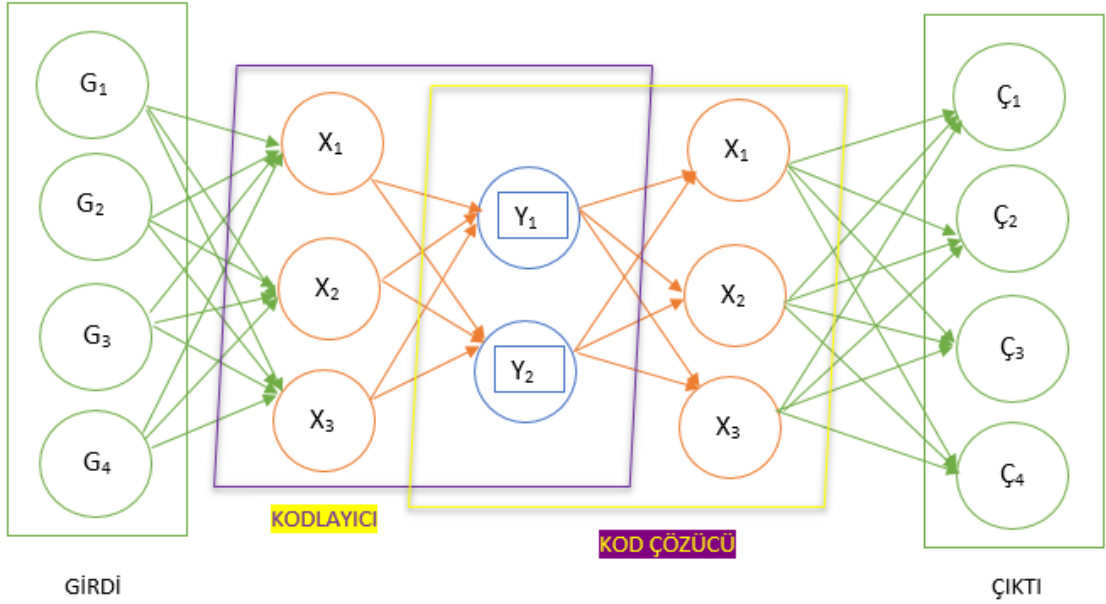
Bir otokodlayıcının çalışma mantığı girdi verisinin çıkışa kopyalanmasıdır. Algoritma içinde girdiyi gösteren bir gizli katman bulunur. Bu sinir ağında iki parça bulunmaktadır. Birinci kısım $b = f(o)$ işlevini sağlayan kodlayıcı, ikinci kısım ise $a = g(b)$ işlevini sağlayan kod çözücüdür. Şekil 4.6'da otokodlayıcı ağının genel yapısı



Şekil 4.6 Otokodlayıcının genel yapısı

görülmektedir. Bu ağ, her yerde $g(f(o)) = a$ eşitliğini sağlarsa bu durum elverişli olmaz. Otokodlayıcılar genellikle aynı veriyi kopyalama işlemini öğrenmeye zorlanır. Yapılan işlem, giriş verisinin yaklaşık değerini çıkışa yansıtma ve sadece eğitim veri seti üzerinde kopyalama işlemini öğrenirler.

Şekil 4.7'de örnek bir otokodlayıcı modeli görülmektedir.



Şekil 4.7 Örnek bir otokodlayıcı modeli

Son 10 yıldır kullanılan otokodlayıcı algoritmaları deterministik değil stokastik fonksiyonlarla uğraşır. Bu nedenle deterministik fonksiyonlar gizli katman ikilisi (kodlayıcı ve kod çözücü) fonksiyonları stokastik fonksiyonlara çevirirler.

Otokodlayıcılar uzun süredir ([88], [89]) kullanılan bir algoritmadır. Genellikle boyut azaltma ve özneteliklerin öğrenimi için kullanılmıştır. Bu ağ, ileri beslemeli ağların özel bir çeşididir. Geriye yayılım tarafından hesaplanan gradyanları izleyen miniyığın gradyan iniş gibi algoritmalarla öğrenirler. Otokodlayıcı türlerine bakacak olursak:

4.5.1 Tamamlanmamış Otokodlayıcı

Bu aşda, girdinin çıktıya yani kod çözücünün çıktıya kopyalanması anlamsız gibi görünebilir fakat burada yapılan işlem kodlayıcı kısmını eğiterek (girdiyi kopyalayarak) sonuçta elde edilen b değişkeninin önemli nitelikler çıkarmasını beklemektir.

Otokodlayıcının bunu yapabilmesi için b değişkeninin (kodlayıcının çıkışı), \mathbf{o} giriş verisinden daha küçük boyutlu olması gerekmektedir [90]. Bu tarz kodlayıcılara tamamlanmamış otokodlayıcı (undercomplete autoencoder) denir. Tamamlanmamış bir veriyi öğrenmek, otokodlayıcının eğitim verisinin en önemli niteliklerini çıkarır. Kodlayıcı ve kodçözücü çok yüksek kapasiteli olduğunda ise önemli bilgileri öğrenemezler. Bilindiği gibi öğrenme süreci, kayıp işlevinin minimize edilmesini amaçlar.

$$P(\mathbf{o}, g(f(\mathbf{o}))), \quad (4.1)$$

Burada P fonksiyonu; $g(f(\mathbf{o}))$ fonksiyonu ile \mathbf{o} girdisi arasındaki farka göre cezalandırılan karesel ortalama hatası gibi bir kayıp işlevidir. Eğrisel bir f kodlayıcı ve g kodçözücüsü içeren bir otokodlayıcı temel bileşenler analizi yönteminin daha güçlü bir gösterimini öğrenir.

4.5.2 Düzenleştirilmiş Otokodlayıcı

Gizli kodun boyutu girdinin boyutuyla eş veya girdinin boyutundan büyükse yani aşırı tamamlanmış (overcomplete) durumunda kod çözücüye kopyalama işlemi, önemli bilgiler öğrenilmeden yapılır.

Kodlayıcı ve kod çözücünün kapasitesi ve kod boyutları modelin yapısına göre değerlendirilirse otokodlayıcı bu durumda başarılı olur. Düzenleştirilmiş otokodlayıcılar bu imkanı sağlar. Yapılan iş; modelin yapısını küçültmek yerine, bir kayıp işlevi yardımıyla model, temel özellik olan girdiyi çıktıya kopyalamanın yanında, başka nitelikler de öğrenir. Örneğin; modelin seyrekliği, türevinin küçüklüğü, gürültülü veya eksik girdiler karşısında sağlamlığı bunlara örnektir.

4.5.2.1 Seyrek Otokodlayıcı

Seyrek otokodlayıcıda kayıp işlevi şu şekildedir:

$$P(\mathbf{o}, g(f(\mathbf{o}))) + \Gamma(b). \quad (4.2)$$

Kodlayıcı çıktısı $b = f(\mathbf{o})$, kod çözücü ise $g(b)$ 'dir. Burada kayıp işlevinde seyreklik payı $\Gamma(b)$ bulunur.

$\Gamma(b)$ seyreklik ceza payı ileri beslemeli bir ağın kayıp işlevine eklenmiş bir düzenleştirici terim olarak tanımlanabilir.

Bu ağ, özellikle sınıflandırma gibi işlerde kullanılan öznitelikleri öğrenmek için kullanılır. Seyrek olması için düzenleştirilmiş olan bu ağ, kopyalama işlemini bir seyreklik ceza payı ile gerçekleştirdiğinde, fazladan sistem için gerekli öznitelikleri öğrenmiş olur.

4.5.2.2 Arıtan Otokodlayıcılar

Kayıp işlevine seyrek otokodlayıcıda olduğu gibi Γ ceza payı vermekten ziyade, bu işlevin yeniden yapılandırma hatası terimini düzenleyerek kullanışlı bilgileri çıkaran bir otokodlayıcı modelidir (denoising autoencoder - DAE).

$$P(\mathbf{o}, g(f(\mathbf{o}))), \quad (4.3)$$

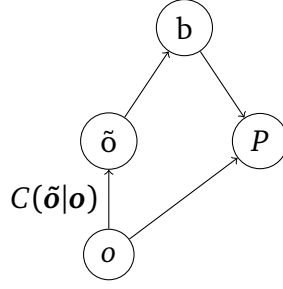
yerine

$$P(\mathbf{o}, g(f(\tilde{\mathbf{o}}))). \quad (4.4)$$

işlevini minimize eder. Burada $\tilde{\mathbf{o}}$ terimi, \mathbf{o} giriş verisinin bir nevi bozucu eklenmiş halidir. Bu sebeple arıtan otokodlayıcının esas görevi olan girdiyi çıktıya kopyalamakla beraber, bu bozulmayı düzeltmesi beklenir.

f ve g kodlayıcı ve kodçözücü güçler [91] ve [92] çalışmalarında olduğu gibi $p_{veri}(\mathbf{o})$ işlevinin yapısını dolaylı yoldan öğrenirler. Bir arıtan otokodlayıcı eğitim süreci şekil 4.8'de görülmektedir.

$C(\tilde{\mathbf{o}}|\mathbf{o})$ biçiminde, bir \mathbf{o} verisinden oluşan bozulmuş $\tilde{\mathbf{o}}$ örnekleri üzerine bir koşullu dağılımı ifade eden bir bozulma süreci tanımlanır. Ardından ağ, $(\mathbf{o}, \tilde{\mathbf{o}})$ eğitim çiftini kullanarak hesaplanan ve $p_{yeniyapı}(\mathbf{o}|\tilde{\mathbf{o}})$ ile tanımlı bir yeniden yapılandırma dağılımını



Şekil 4.8 Arıtan otokodlayıcının genel yapısı

ezberler. Süreç şöyle işler:

- 1) Eğitim verisinden alınan bir \mathbf{o} eğitim örneği oluştur.
- 2) $C(\tilde{\mathbf{o}}|\mathbf{o})$ işlevinden, bozulmuş bir ($\tilde{\mathbf{o}}$) elde edilir.
- 3) $(\mathbf{o}, \tilde{\mathbf{o}})$ çifti $p_{yeni yapı}(\mathbf{o}|\tilde{\mathbf{o}}) = p_{kodcozucu}(\mathbf{o}|\mathbf{b})$ otokodlayıcının bu dağılmış işlevi hesaplamak için eğitimde kullanılır. \mathbf{b} ; $f(\tilde{\mathbf{o}})$ kodlayıcısının çıktısı, $p_{kodcozucu}$ ise $g(\mathbf{b})$ 'dir.

4.5.2.3 Türevleri Cezalandırarak Düzenleştirme

Otokodlayıcıları düzenleştirmenin yöntemi kayıp işlevine bir ceza payı eklemektir. Bu ağda ceza payı şu şekilde ifade edilir:

$$P(\mathbf{o}, g(f(\mathbf{o}))) + \Gamma(\mathbf{b}, \mathbf{o}), \quad (4.5)$$

Γ ceza payı burada şu formdadır:

$$\Gamma(\mathbf{b}, \mathbf{o}) = \lambda \sum_i \|\nabla_{\mathbf{o}} b_i\|^2. \quad (4.6)$$

Bu terim, modeli \mathbf{o} giriş verisinin fazla olmayan değişimlerine karşı sağlam olmayı öğrenmeye zorlar. Burada ceza payı, sadece eğitim veri setini kapsar ve ağı eğitim dağılımı hakkında yararlı bilgileri saklayan öznelikleri ezberlemeye iter. Bu ağ şekline daraltan otokodlayıcı (contractive autoencoder-CAE) denir.

4.6 Deneysel Sonuçlar

Bu bölümde önerilen imge kaynaştırma modeli olan arıtan otokodlayıcı modeli için kodlayıcı kısmında evrişim katmanları olarak Conv2D(5, 5)+ MaxPooling2D(2, 2) kullanılmıştır. Kod çözücü kısmında ise evrişim katmanlarını Conv2D(5, 5)+ UpSampling2D(2, 2) oluşturmaktadır. Şekil 4.9'da arıtan otokodlayıcı modelinin kodlayıcı kısmı, Şekil 4.10'da ise otokodlayıcı modelinin kod çözücü kısmı

görülmektedir. Çakıştırılmış imgelerden 80 adet imge eğitim veri seti olarak , 20 adet imge ise test veri seti olarak ayrılmıştır. Optimizasyon algoritması olarak Adaptif Momentum, kayıp işlevi olarak da binary crossentropy kullanılmıştır. Şekil 4.9 ve Şekil 4.10'da kullanılan filtre sayıları, aktivasyon fonksiyonları ve kenar etkisi seçimleri belirtilmiştir.

```

TR=np.expand_dims(TR,axis=-1)
Giris = Input(shape=(512, 512, 1))

#kodlayici mimarisi
x1 = Conv2D(128, (5,5), activation='relu', padding='same')(Giris)
x1 = MaxPool2D( (2, 2), padding='same')(x1)
x2 = Conv2D(128, (5,5), activation='relu', padding='same')(x1)
x2 = MaxPool2D( (2, 2), padding='same')(x2)
x3 = Conv2D(128, (5,5), activation='relu', padding='same')(x2)
x3 = MaxPool2D( (2, 2), padding='same')(x3)
x4 = Conv2D(64, (5,5), activation='relu', padding='same')(x3)
x4 = MaxPool2D( (2, 2), padding='same')(x4)
x5 = Conv2D(64, (5,5), activation='relu', padding='same')(x4)
x5 = MaxPool2D( (2, 2), padding='same')(x5)
x6 = Conv2D(32, (5,5), activation='relu', padding='same')(x5)
x6 = MaxPool2D( (2, 2), padding='same')(x6)
x7 = Conv2D(32, (5,5), activation='relu', padding='same')(x6)
kodlayici = MaxPool2D( (2, 2), padding='same')(x7)

```

Şekil 4.9 Önerilen kodlayıcı mimarisi

```

# kodcozucu mimarisi
x7 = Conv2D(32, (5,5), activation='relu', padding='same')(kodlayici)
x7 = UpSampling2D((2, 2))(x7)
x6 = Conv2D(32, (5,5), activation='relu', padding='same')(x7)
x6 = UpSampling2D((2, 2))(x6)
x5 = Conv2D(64, (5,5), activation='relu', padding='same')(x6)
x5 = UpSampling2D((2, 2))(x5)
x4 = Conv2D(64, (5,5), activation='relu', padding='same')(x5)
x4 = UpSampling2D((2, 2))(x4)
x3 = Conv2D(128, (5,5), activation='relu', padding='same')(x4)
x3 = UpSampling2D((2, 2))(x3)
x2 = Conv2D(128, (5,5), activation='relu', padding='same')(x3)
x2 = UpSampling2D((2, 2))(x2)
x1 = Conv2D(128, (5,5), activation='relu', padding='same')(x2)
x1 = UpSampling2D((2, 2))(x1)
kodcozucu = Conv2D(1, (5,5), padding='same')(x1)

```

Şekil 4.10 Önerilen kod çözücü mimarisi

Bu bölümde, SIFT-Akış yöntemiyle elde edilen belirli çakıştırılmış veri setleri üzerine

(veri seti 1, veri seti 2, veri seti 4, veri seti 15, veri seti 17 ve Gilles'in [3] çalışmasındaki "door" veri seti) LAPLACE tabanlı imge kaynaştırma, DTCWT (Çift ağaç karmaşık dalgacık dönüşümü)[5] tabanlı imge kaynaştırma, Temel bileşenler analizi (TBA) tabanlı imge kaynaştırma ve önerdiğimiz evrişimli sinir ağları tabanlı arıtan otokodlayıcı modeli imge kaynaştırma yöntemleri uygulanmış ve sonuçları gösterilmiştir. Şekil 4.11'den Şekil 4.22'ye kadar olan imgeler yöntemlerin görsel karşılaştırmalarını içermektedir.

Gösterilen şekillerde Şekil 4.11 veri seti 1, Şekil 4.13 veri seti 2, Şekil 4.15 veri seti 4, Şekil 4.17 veri seti 15, Şekil 4.19 veri seti 17 ve Şekil 4.21 de door[3] veri seti olmak üzere her bir veri seti için (a) ile temsil edilen imge bozulmuş imgeyi, (b) ile temsil edilen imge SIFT-Akış yöntemi sonucunda oluşturulan çakıştırılmış imgeyi, (c) ile temsil edilen imge LAPLACE tabanlı imge kaynaştırma sonucunu, (d) ile temsil edilen imge TBA tabanlı imge kaynaştırma sonucunu, (e) ile temsil edilen imge DTCWT [5] tabanlı imge kaynaştırma sonucunu ve (f) ile temsil edilen imge ise bu çalışma için geliştirilen arıtan otokodlayıcı modeli imge kaynaştırma sonucunu göstermektedir.

Tablo 4.1, Tablo 4.2 ve Tablo 4.3 sırasıyla yapısal benzerlik indis ölçümü, tepe sinyal gürültü oranı ve ilinti katsayısı sonuçlarını göstermektedir ve bu nesnel metrik sonuçlarına göre en etkin performansı, önerilen arıtan otokodlayıcı modeli göstermektedir.

Şekil 4.12, 4.14, 4.16, 4.18, 4.20, 4.22 ise sırasıyla her bir veri seti için ilgilenilen bölgelerin (alt bölge) kaynaştırma sonuçlarını göstermektedir. Görsel sonuçlar da nesnel ölçüm metrikleri gibi önerilen arıtan otokodlayıcı yönteminin en iyi sonucu çıkardığını göstermektedir.

Tablo 4.1 Yapısal Benzerlik Ölçümü Metriği (SSIM) Sonuçları

| İmge Seti | SSIM | | | |
|---------------------|---------|--------|-----------|---------------|
| | Laplace | TBA | DTCWT [5] | Önerilen |
| Veri Seti-1 | 0.9260 | 0.9342 | 0.8936 | 0.9701 |
| Veri Seti-2 | 0.9099 | 0.8534 | 0.8461 | 0.9668 |
| Veri Seti-4 | 0.8834 | 0.8025 | 0.8518 | 0.9651 |
| Veri Seti-15 | 0.9179 | 0.8592 | 0.8470 | 0.9497 |
| Veri Seti-17 | 0.9139 | 0.9051 | 0.8872 | 0.9528 |
| door [3] | 0.8533 | 0.8163 | 0.8069 | 0.9078 |

Tablo 4.2 Tepe Sinyal Gürültü Oranı (PSNR) Metriği Sonuçları

| İmge Seti | PSNR | | | |
|--------------|---------|-------|-----------|--------------|
| | Laplace | TBA | DTCWT [5] | Önerilen |
| Veri Seti-1 | 35.99 | 34.79 | 29.72 | 39.38 |
| Veri Seti-2 | 29.98 | 26.63 | 28.26 | 36.40 |
| Veri Seti-4 | 25.27 | 22.92 | 25.54 | 29.96 |
| Veri Seti-15 | 30.93 | 27.16 | 28.44 | 35.48 |
| Veri Seti-17 | 38.28 | 35.30 | 36.92 | 41.27 |
| door [3] | 27.65 | 24.01 | 27.85 | 31.09 |



Tablo 4.3 İlinti Katsayısı (NCC) Metriği Sonuçları

| İmge Seti | NCC | | | |
|--------------|---------|--------|-----------|---------------|
| | Laplace | TBA | DTCWT [5] | Önerilen |
| Veri Seti-1 | 0.9843 | 0.9822 | 0.9716 | 0.9897 |
| Veri Seti-2 | 0.9788 | 0.9531 | 0.9648 | 0.9961 |
| Veri Seti-4 | 0.9535 | 0.9157 | 0.9414 | 0.9784 |
| Veri Seti-15 | 0.9727 | 0.9360 | 0.9461 | 0.9871 |
| Veri Seti-17 | 0.9898 | 0.9879 | 0.9843 | 0.9945 |
| door [3] | 0.9684 | 0.9369 | 0.9282 | 0.9859 |



(a) Bozulmuş imge



(b) akıştırılmış imge



(c) LAPLACE tabanlı imge kaynaştırma sonucu



(d) TBA tabanlı imge kaynaştırma sonucu

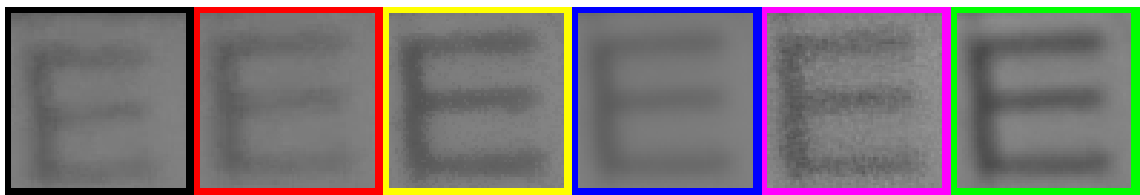


(e) DTCWT [5] imge kaynaştırma sonucu



(f) Evrişimli sinir ağı imge kaynaştırma sonucu

Şekil 4.11 Veri seti-1 için imge kaynaştırma sonuçları



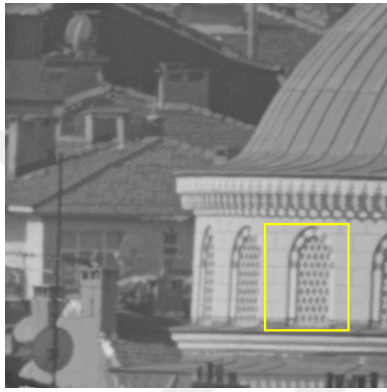
Şekil 4.12 Veri seti-1 için örnek bir alt imge sonucu



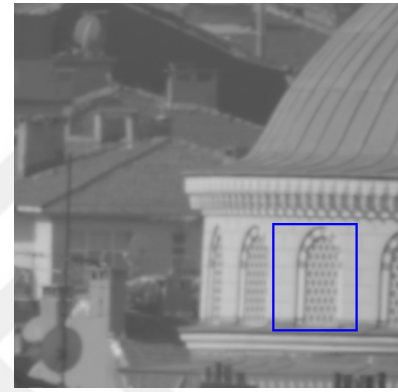
(a) Bozulmuş imge



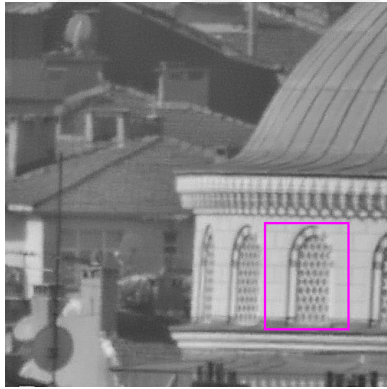
(b) akıřtırılmıř imge



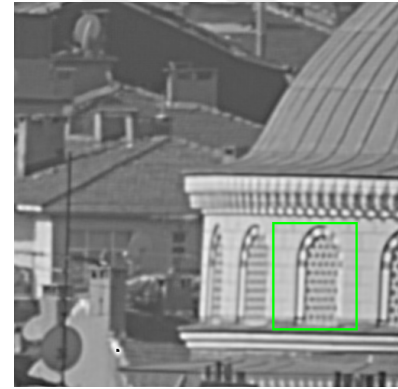
(c) LAPLACE tabanlı imge kaynařtırma sonucu



(d) TBA tabanlı imge kaynařtırma sonucu

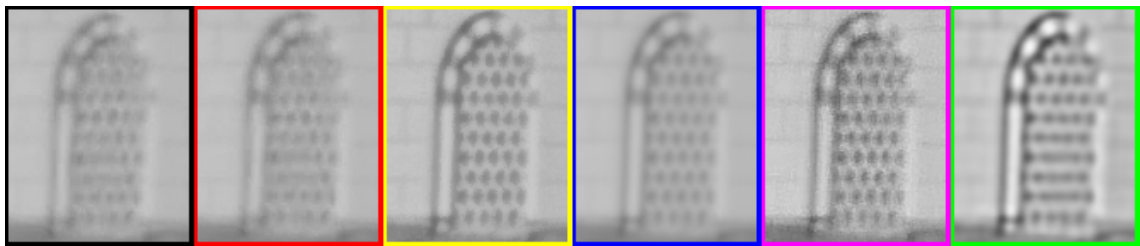


(e) DTCWT [5] imge kaynařtırma sonucu



(f) Evriřimli sinir ađı imge kaynařtırma sonucu

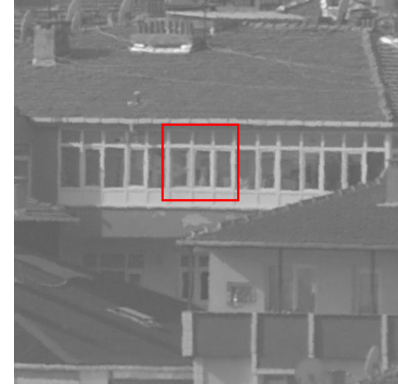
řekil 4.13 Veri seti-2 iin imge kaynařtırma sonuları



řekil 4.14 Veri seti-2 iin rnek bir alt imge sonucu



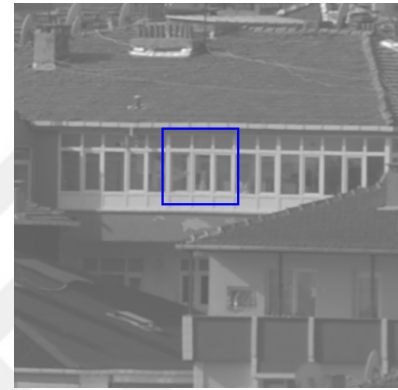
(a) Bozulmuş imge



(b) akıřtırılmıř imge



(c) LAPLACE tabanlı imge kaynařtırma sonucu



(d) TBA tabanlı imge kaynařtırma sonucu

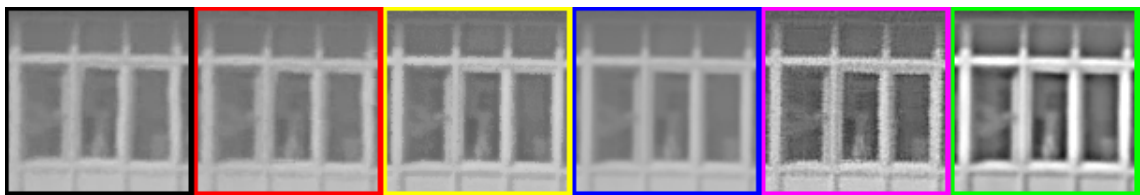


(e) DTCWT [5] imge kaynařtırma sonucu



(f) Evriřimli sinir ađı imge kaynařtırma sonucu

řekil 4.15 Veri seti-4 iin imge kaynařtırma sonuları



řekil 4.16 Veri seti-4 iin rnek bir alt imge sonucu



(a) Bozulmuş imge



(b) Çakıştırılmış imge



(c) LAPLACE tabanlı imge kaynaştırma sonucu



(d) TBA tabanlı imge kaynaştırma sonucu

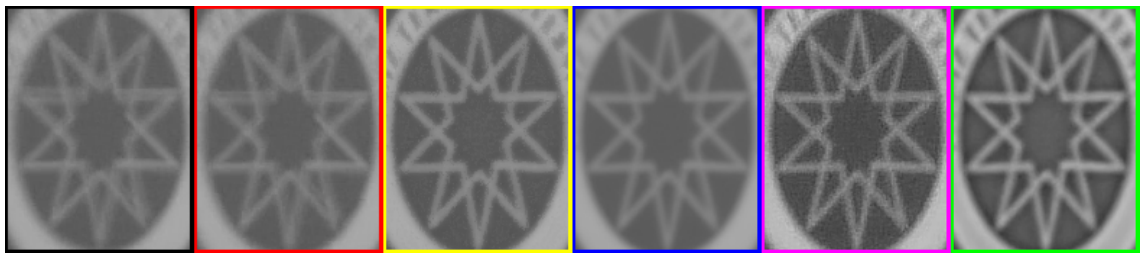


(e) DTCWT [5] imge kaynaştırma sonucu

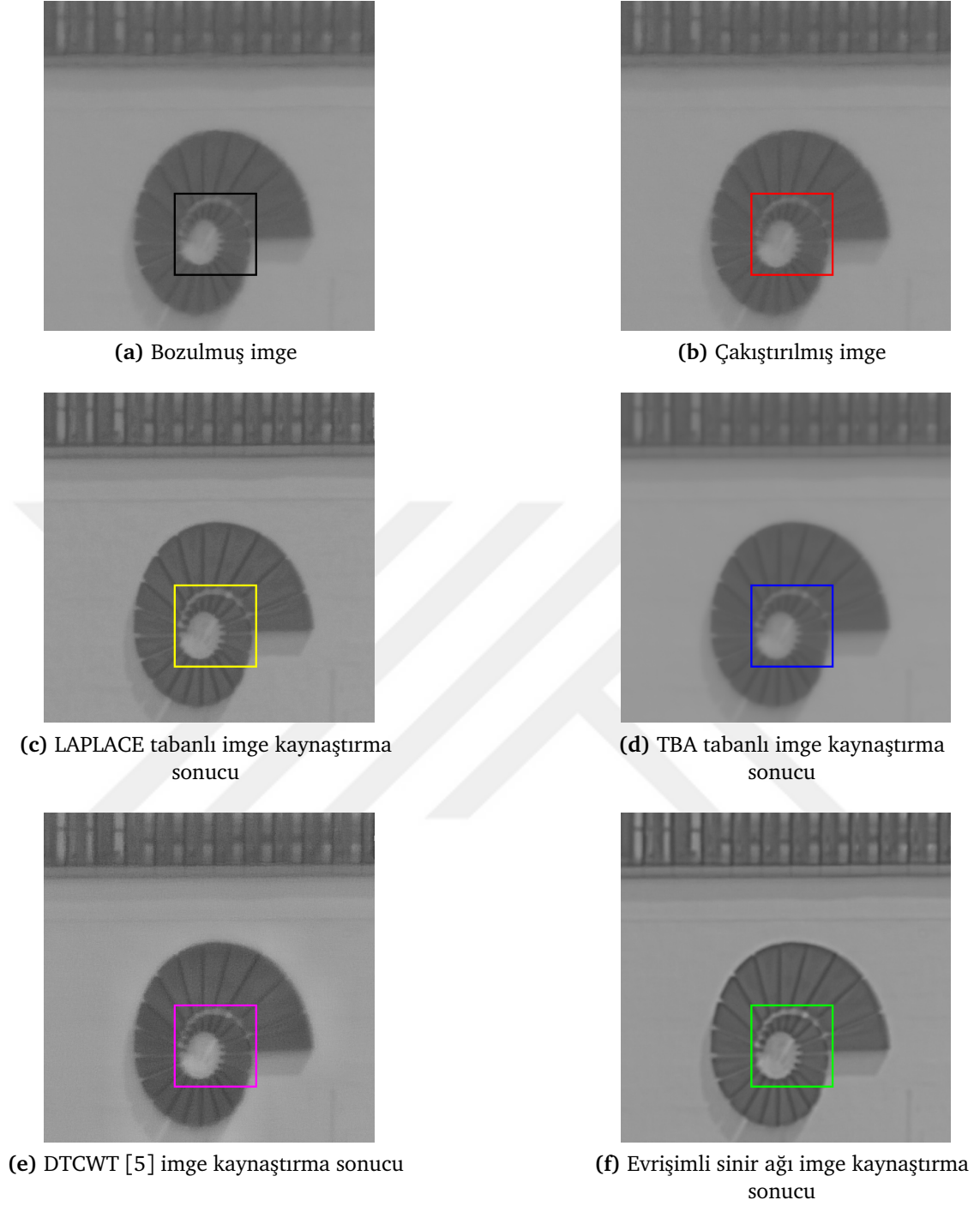


(f) Evrişimli sinir ağı imge kaynaştırma sonucu

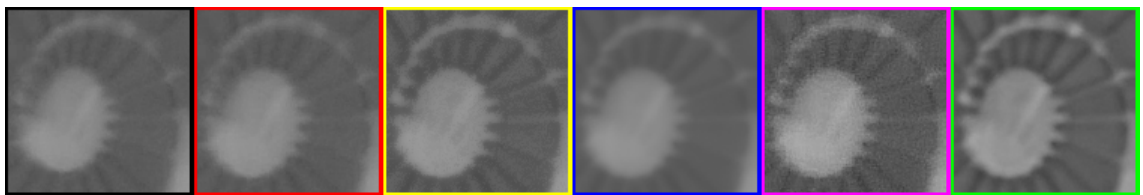
Şekil 4.17 Veri seti-15 için imge kaynaştırma sonuçları



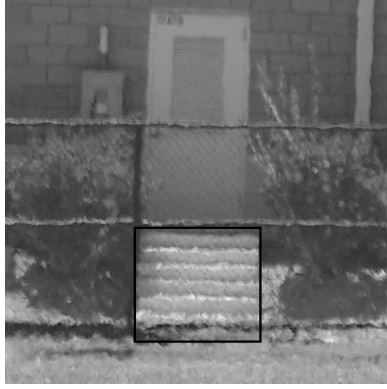
Şekil 4.18 Veri seti-15 için örnek bir alt imge sonucu



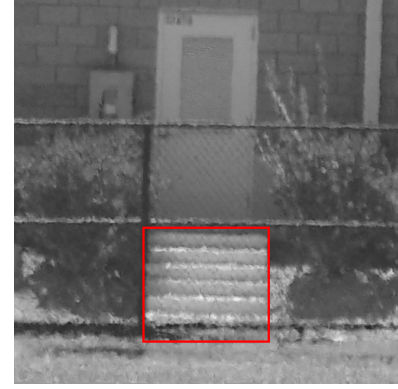
Şekil 4.19 Veri seti-15 için imge kaynaştırma sonuçları



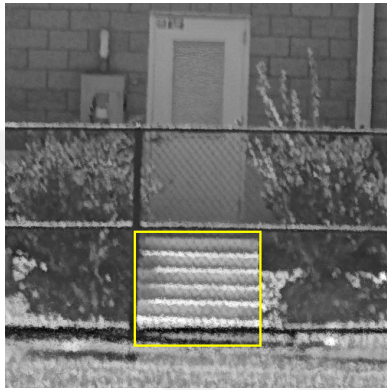
Şekil 4.20 Veri seti-17 için örnek bir alt imge sonucu



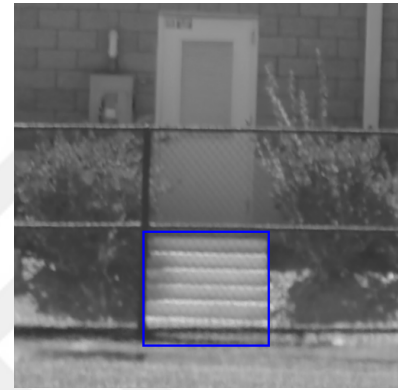
(a) Bozulmuş imge



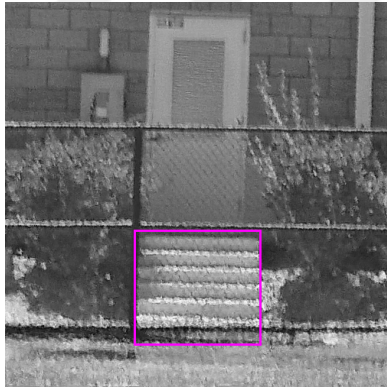
(b) akıştırılmış imge



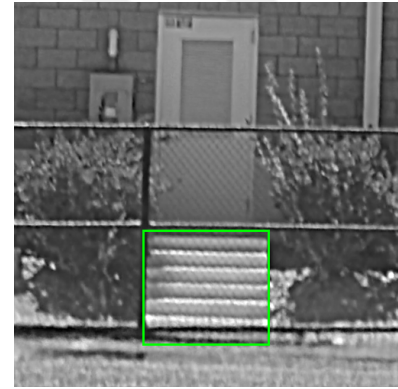
(c) LAPLACE tabanlı imge kaynaştırma sonucu



(d) TBA tabanlı imge kaynaştırma sonucu

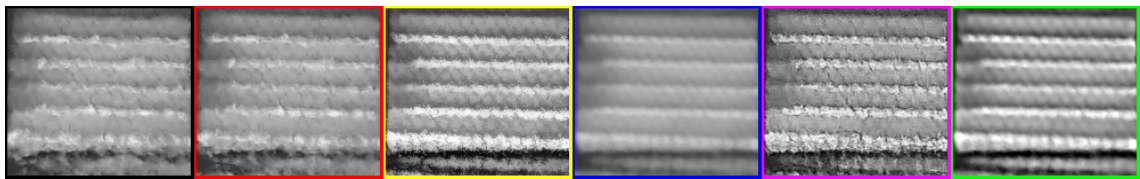


(e) DTCWT [5] imge kaynaştırma sonucu



(f) Evrişimli sinir ağı imge kaynaştırma sonucu

Şekil 4.21 Veri seti-door [3] için imge kaynaştırma sonuçları



Şekil 4.22 Veri seti-door [3] için örnek bir alt imge sonucu

5 SONUÇ VE ÖNERİLER

Uzak mesafeli görüntülemelerde imgelerin makinalar veya insanlar tarafından tanınabilmesi, tespit edilebilmesi veya takip edilebilmesi için zorlu ortam koşullarında, geniş alanlarda ve/veya insanların ulaşamayacağı yerlerde gözlenen sahnenin elde edilen imgelerinde, bozulmaların veya geometrik deformasyonların giderilmesine ihtiyaç vardır.

Atmosferik türbülans kaynaklı bozulmalar; uzak mesafeli görüntülemelerde duruk imgelerde bulunan baskın bozulma türüdür. Bu etkiden kaynaklı imge bozulmalarında, hareket veya odak dışı bulanıklığı gibi tekdüze bir bozulmadan bahsedilemez. Bu nedenle imgenin ya alt parçalara bölünüp tekdüze bulanıklık giderici filtrelerle iyileştirilmesi ya da imgenin aynı sahne olmak kaydıyla imge dizisine ihtiyaç duyulur.

İmgelerde atmosferik türbülans etkilerinin azaltılmasına yönelik çalışmalarda veri setine ihtiyaç vardır. Fakat halihazırda bu konuda açık kaynak kullanılabilecek yeterli imge seti mevcut değildir. Bu alanda var olan açık kaynak veri seti (bireysel araştırmacılara yönelik) eksikliğine alternatif bir çözüm sunulmuştur.

Atmosferik türbülans bozulmaları imgede yerel dalgalanmalara ve dolayısıyla da nesnelere geometrik deformasyonlara neden olduğundan bozulma modelinin çıkarılması zorlu bir görevdir. Bu nedenle bu çalışmanın 2. bölümünde veri modellerinden GKM ve PDM yöntemleriyle imge dizilerinde bulunan yerel bozulmaların imgenin hangi bölgelerinde olduğu görsel olarak temsil edilmiştir.

Elde edilmiş ham imgelerin geometrik deformasyonları azaltılmış tek bir imgeye indirgenebilmesi için öncelikli işlem, belirlenecek bir referans imgeye göre karşılıklı noktaları denk gelecek şekilde çakıştırılmalarıdır. 3. bölümde imge dizilerine katı olmayan imge çakıştırma yöntemlerinden üç farklı algoritma uygulanmış ve bu sonuçların etkinliği karşılaştırılmıştır.

İmge çakıştırması sonrası elde edilen çakıştırılmış imge dizisinin, geometrik

deformasyonu ve bulanıklığı azaltılmış tek bir imgeye kaynaştırılması için evrişimli sinir ağıları tabanlı bir yöntem önerilmiş ve sonuçları literatürde bulunan bazı imge kaynaştırma yöntemleriyle birlikte sunulmuştur.

İmge çıkıştırma işleminde düzenlilik teriminin önemi fazla olduğundan bundan sonraki çalışmalarda bu konuya ve optimizasyon stratejilerine ağırlık verilecektir. Yapılması hedeflenen bir diğer çalışma da, türbülans bozulmalarını azaltmak için önerilen yöntemin performansını iyileştirmeye yönelik olacaktır.



-
- [1] T. Çalışkan N. Arica, "Atmospheric turbulence mitigation using feature based optical flow," in *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2014, pp. 1750–1753.
- [2] X. Zhu P. Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 157–170, Jan. 2013, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.82. [Online]. Available: <https://doi.org/10.1109/TPAMI.2012.82>.
- [3] J. Gilles N. B. Ferrante, "Open turbulent image set (otıs)," *Pattern Recognition Letters*, vol. 86, pp. 38–41, 2017.
- [4] N. Anantrasirichai, A. Achim, D. Bull, "Mitigating the effects of atmospheric distortion," 2012.
- [5] N. Anantrasirichai, A. Achim, N. G. Kingsbury, D. R. Bull, "Atmospheric turbulence mitigation using complex wavelet-based fusion," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2398–2408, 2013.
- [6] C. Carrano J. Brase, "Horizontal and slant-path surveillance with speckle imaging," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2002.
- [7] X. Zhu P. Milanfar, "Stabilizing and deblurring atmospheric turbulence," in *2011 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2011, pp. 1–8.
- [8] J. Gilles, T. Dagobert, C. De Franchis, "Atmospheric turbulence restoration by diffeomorphic image registration and blind deconvolution," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer, 2008, pp. 400–409.
- [9] J. P. Bos M. C. Roggemann, "Blind image quality metrics for optimal speckle image reconstruction in horizontal imaging scenarios," *Optical Engineering*, vol. 51, no. 10, p. 107 003, 2012.
- [10] M. Hirsch, S. Sra, B. Schölkopf, S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 607–614.
- [11] R. Hufnagel N. Stanley, "Modulation transfer function associated with image transmission through turbulent media," *JOSA*, vol. 54, no. 1, pp. 52–61, 1964.
- [12] D. Li, R. M. Mersereau, D. H. Frakes, M. J. Smith, "A new method for suppressing optical turbulence in video," in *2005 13th European Signal Processing Conference*, IEEE, 2005, pp. 1–4.

- [13] D. Li S. Simske, "Atmospheric turbulence degraded-image restoration by kurtosis minimization," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 2, pp. 244–247, 2009.
- [14] D. L. Fried, "Probability of getting a lucky short-exposure image through turbulence," *JOSA*, vol. 68, no. 12, pp. 1651–1658, 1978.
- [15] J. Gilles S. Osher, "Fried deconvolution," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIII*, International Society for Optics and Photonics, vol. 8355, 2012, 83550G.
- [16] C.-A. Deledalle J. Gilles, "Batud: Blind atmospheric turbulence deconvolution," 2019.
- [17] C. S. Huebner S. Gladysz, "Simulation of atmospheric turbulence for a qualitative evaluation of image restoration algorithms with motion detection," in *Optics in Atmospheric Propagation and Adaptive Systems XV*, International Society for Optics and Photonics, vol. 8535, 2012, p. 85350L.
- [18] B. L. McGlamery, "Computer simulation studies of compensation of turbulence degraded images," in *Image processing*, International Society for Optics and Photonics, vol. 74, 1976, pp. 225–233.
- [19] E. Repasi R. Weiss, "Analysis of image distortions by atmospheric turbulence and computer simulation of turbulence effects," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIX*, International Society for Optics and Photonics, vol. 6941, 2008, 69410S.
- [20] A. Schwartzman, M. Alterman, R. Zamir, Y. Y. Schechner, "Turbulence-induced 2d correlated image distortion," in *2017 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2017, pp. 1–13.
- [21] J. P. Bos M. C. Roggemann, "Technique for simulating anisoplanatic image formation over long horizontal paths," *Optical Engineering*, vol. 51, no. 10, p. 101 704, 2012.
- [22] Y. Li, Y. Iwamoto, K. Ogawa, Y.-W. Chen, "Computer simulation of image distortion by atmospheric turbulence using time-series image data with 250-million-pixels," in *International Journal of Computer and Electrical Engineering*, press, 2017.
- [23] N. Chimitt, Z. Mao, G. Hong, S. H. Chan, "Rethinking atmospheric turbulence mitigation," *arXiv preprint arXiv:1905.07498*, 2019.
- [24] A. Labeyrie, "Attainment of diffraction limited resolution in large telescopes by fourier analysing speckle patterns in star images," *Astron. Astrophys.*, vol. 6, no. 1, pp. 85–87, 1970.
- [25] C. J. Carrano, "Speckle imaging over horizontal paths," in *High-Resolution Wavefront Control: Methods, Devices, and Applications IV*, International Society for Optics and Photonics, vol. 4825, 2002, pp. 109–120.
- [26] C. J. Carrano, "Progress in horizontal and slant-path imaging using speckle imaging," in *Optical Engineering at the Lawrence Livermore National Laboratory*, T. T. Saito M. A. Lane, Eds., International Society for Optics and Photonics, vol. 5001, SPIE, 2003, pp. 56–64. DOI: 10 . 1117 / 12 . 500368. [Online]. Available: <https://doi.org/10.1117/12.500368>.

- [27] C. Carrano, "Bispectral speckle imaging algorithm performance on specific simulated scenarios," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2013.
- [28] M. R. S. Jahromi, A. Lambert, D. Fraser, G. Thorpe, D. Clyde, J. Webb, M. Tahtali, "Image restoration of images obtained by near-horizontal imaging through the atmosphere," *DICTA2002: Digital Image Computing Techniques and Applications*, vol. 1, pp. 21–22, 2002.
- [29] Z. Wen, D. Fraser, A. Lambert, "Bicoherence used to predict lucky regions in turbulence affected surveillance," in *2006 IEEE International Conference on Video and Signal Based Surveillance*, IEEE, 2006, pp. 108–108.
- [30] Y. Lou, S. H. Kang, S. Soatto, A. L. Bertozzi, "Video stabilization of atmospheric turbulence distortion," *Inverse Probl. Imaging*, vol. 7, no. 3, pp. 839–861, 2013.
- [31] A. Patel, D. Israni, N. A. M. Kumar, C. Bhatt, "An adaptive image registration technique to remove atmospheric turbulence," *Statistics, Optimization & Information Computing*, vol. 7, no. 2, p. 439, 2019.
- [32] G. Gezer M. Mercimek, "An investigation of registration methods on the images degraded by atmospheric turbulence," in *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, IEEE, 2018, pp. 1–5.
- [33] R. C. Hardie, M. A. Rucci, A. J. Dapore, B. K. Karch, "Block matching and wiener filtering approach to optical turbulence mitigation and its application to simulated and real imagery with quantitative error analysis," *Optical Engineering*, vol. 56, no. 7, p. 071 503, 2017.
- [34] M. A. Rucci, R. C. Hardie, A. J. Dapore, "Comparing multiple turbulence restoration algorithms performance on noisy anisoplanatic imagery," in *Long-Range Imaging II*, International Society for Optics and Photonics, vol. 10204, 2017, p. 1 020 409.
- [35] J. Flusser, F. Sroubek, B. Zitova, "Image fusion: Principles, methods, and applications," *Tutorial Eusipco*, 2007.
- [36] M. Shimizu, S. Yoshimura, M. Tanaka, M. Okutomi, "Super-resolution from image sequence under influence of hot-air optical turbulence," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Citeseer, 2008, pp. 1–8.
- [37] A. Lambert, F. Li, D. Bowman, D. Fraser, "Experiments in turbulence induced super-resolution in surveillance imagery," in *28th International Congress on High-Speed Imaging and Photonics*, International Society for Optics and Photonics, vol. 7126, 2009, p. 712 612.
- [38] R. Nieuwenhuizen, J. Dijk, K. Schutte, "Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects," *EURASIP journal on image and video processing*, vol. 2019, no. 1, p. 2, 2019.
- [39] Z. Bin, Q. Ying-jian, O. Zhi-wei, J. Yan-li, "Regularized image restoration with singular value decomposition," in *2011 International Conference on Image Analysis and Signal Processing*, IEEE, 2011, pp. 321–325.
- [40] R. C. Gonzales R. E. Woods, *Digital image processing*, 2002.

- [41] A. N. Kolmogorov, "The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers," *Cr Acad. Sci. URSS*, vol. 30, pp. 301–305, 1941.
- [42] A. Oboukhov, "Structure of the temperature field in turbulent flows," *Isv. Geogr. Geophys. Ser.*, vol. 13, pp. 58–69, 1949.
- [43] S. Corrsin, "On the spectrum of isotropic temperature fluctuations in an isotropic turbulence," *Journal of Applied Physics*, vol. 22, no. 4, pp. 469–473, 1951.
- [44] A. Ishimaru, *Wave propagation and scattering in random media*. Academic press New York, 1978, vol. 2.
- [45] J. Schmidt, "Numerical simulation of optical wave propagation with examples in matlab," Society of Photo-Optical Instrumentation Engineers, 2010.
- [46] T. Bouwmans, F. El Baf, B. Vachon, "Statistical background modeling for foreground detection: A survey," in *Handbook of pattern recognition and computer vision*, World Scientific, 2010, pp. 181–199.
- [47] N. J. McFarlane C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine vision and applications*, vol. 8, no. 3, pp. 187–193, 1995.
- [48] E. Chen, O. Haik, Y. Yitzhaky, "Detecting and tracking moving objects in long-distance imaging through turbulent medium," *Applied optics*, vol. 53, no. 6, pp. 1181–1190, 2014.
- [49] C. Stauffer W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, IEEE, vol. 2, 1999, pp. 246–252.
- [50] B. Zitova J. Flusser, "Image registration methods: A survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [51] W. R. Crum, T. Hartkens, D. Hill, "Non-rigid image registration: Theory and practice," *The British journal of radiology*, vol. 77, no. suppl_2, S140–S153, 2004.
- [52] M. Holden, "A review of geometric transformations for nonrigid body registration," *IEEE transactions on medical imaging*, vol. 27, no. 1, pp. 111–128, 2007.
- [53] J. V. Hajnal D. L. Hill, *Medical image registration*. CRC press, 2001.
- [54] G. E. Christensen, R. D. Rabbitt, M. I. Miller, "Deformable templates using large deformation kinematics," *IEEE transactions on image processing*, vol. 5, no. 10, pp. 1435–1447, 1996.
- [55] J.-P. Thirion, "Non-rigid matching using demons," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 1996, pp. 245–251.
- [56] B. K. Horn B. G. Schunck, "Determining optical flow," in *Techniques and Applications of Image Understanding*, International Society for Optics and Photonics, vol. 281, 1981, pp. 319–331.

- [57] X. Pennec, P. Cachier, N. Ayache, “Understanding the “demon’s algorithm”:
3d non-rigid registration by gradient descent,” in *International Conference on
Medical Image Computing and Computer-Assisted Intervention*, Springer, 1999,
pp. 597–605.
- [58] H. Wang, L. Dong, J. O’Daniel, R. Mohan, A. S. Garden, K. K. Ang, D. A. Kuban,
M. Bonnen, J. Y. Chang, R. Cheung, “Validation of an accelerated ‘demons’
algorithm for deformable image registration in radiation therapy,” *Physics in
Medicine & Biology*, vol. 50, no. 12, p. 2887, 2005.
- [59] D. G. Lowe *et al.*, “Object recognition from local scale-invariant features.,” in
iccv, vol. 99, 1999, pp. 1150–1157.
- [60] C. Liu, J. Yuen, A. Torralba, “Sift flow: Dense correspondence across scenes and
its applications,” *IEEE transactions on pattern analysis and machine intelligence*,
vol. 33, no. 5, pp. 978–994, 2010.
- [61] F. L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of
deformations,” *IEEE Transactions on pattern analysis and machine intelligence*,
vol. 11, no. 6, pp. 567–585, 1989.
- [62] L. G. Brown, “A survey of image registration techniques,” *ACM computing sur-
veys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.
- [63] J. Kim J. A. Fessler, “Intensity-based image registration using robust correlation
coefficients,” *IEEE transactions on medical imaging*, vol. 23, no. 11, pp. 1430–
1444, 2004.
- [64] C. E. Shannon, “A mathematical theory of communication,” *Bell system technical
journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [65] J. P. Pluim, J. A. Maintz, M. A. Viergever, “Mutual-information-based
registration of medical images: A survey,” *IEEE transactions on medical imaging*,
vol. 22, no. 8, pp. 986–1004, 2003.
- [66] Z. Chen S. Haykin, “On different facets of regularization theory,” *Neural Com-
putation*, vol. 14, no. 12, pp. 2791–2846, 2002.
- [67] A. N. Tikhonov V. Y. Arsenin, “Solutions of ill-posed problems,” *New York*, pp. 1–
30, 1977.
- [68] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear programming: theory and
algorithms*. John Wiley & Sons, 2013.
- [69] D. G. Luenberger, *Introduction to linear and nonlinear programming*.
Addison-Wesley Reading, MA, 1973, vol. 28.
- [70] E. K. Chong S. H. Zak, *An introduction to optimization*. John Wiley & Sons, 2004.
- [71] L. Grippo, F. Lampariello, S. Lucidi, “A nonmonotone line search technique for
newton’s method,” *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–
716, 1986.
- [72] J. J. Moré D. J. Thuente, “Line search algorithms with guaranteed sufficient
decrease,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 20, no. 3,
pp. 286–307, 1994.
- [73] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes
3rd edition: The art of scientific computing*. Cambridge university press, 2007.

- [74] C. Wang, N. Komodakis, N. Paragios, “Markov random field modeling, inference & learning in computer vision & image understanding: A survey,” *Computer Vision and Image Understanding*, vol. 117, no. 11, pp. 1610–1627, 2013.
- [75] Y. Boykov V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.
- [76] T. W. Tang A. C. Chung, “Non-rigid image registration using graph-cuts,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2007, pp. 916–924.
- [77] Q. Yang, L. Wang, N. Ahuja, “A constant-space belief propagation algorithm for stereo matching,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1458–1465.
- [78] A. Hore D. Ziou, “Image quality metrics: Psnr vs. ssm,” in *2010 20th International Conference on Pattern Recognition*, IEEE, 2010, pp. 2366–2369.
- [79] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [80] A. A. Goshtasby S. Nikolov, “Image fusion: Advances in the state of the art,” *Information fusion*, vol. 2, no. 8, pp. 114–118, 2007.
- [81] F. Chollet, *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.
- [82] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber, “Flexible, high performance convolutional neural networks for image classification,” in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [83] A. Krizhevsky, I. Sutskever, G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [84] S. Ioffe C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [85] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [86] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [87] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [88] Y. Lecun, “Phd thesis: Modeles connexionnistes de l’apprentissage (connectionist learning models),” 1987.
- [89] G. E. Hinton R. S. Zemel, “Autoencoders, minimum description length and helmholtz free energy,” in *Advances in neural information processing systems*, 1994, pp. 3–10.

- [90] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*. MIT press, 2016.
- [91] G. Alain Y. Bengio, “What regularized auto-encoders learn from the data-generating distribution,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3563–3593, 2014.
- [92] Y. Bengio, L. Yao, G. Alain, P. Vincent, “Generalized denoising auto-encoders as generative models,” in *Advances in neural information processing systems*, 2013, pp. 899–907.



TEZDEN ÜRETİLMİŞ YAYINLAR

İletişim Bilgisi: gulsungezer@gmail.com

Konferans Bildirisi

1. G. Gezer, M. Mercimek, "An investigation of registration methods on the images degraded by atmospheric turbulence," in 2018 6th International Conference on Control Engineering Information Technology (CEIT), IEEE, 2018, pp. 1–5