

**KOCAELİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ  
ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**HİPERSPEKTRAL GÖRÜNTÜLERİN BÖLÜTLEME VE  
SEYREK GÖSTERİM TEMELLİ SIKIŞTIRILMASI**

**ADEM ERTEM**

**KOCAELİ 2020**

**KOCAELİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

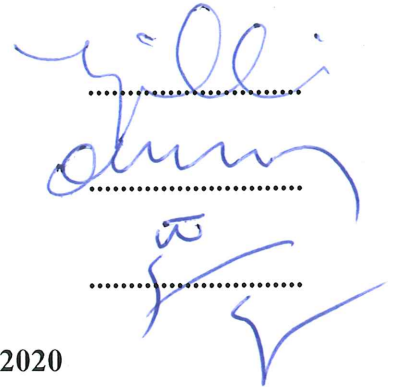
**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ**  
**ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**HİPERSPEKTRAL GÖRÜNTÜLERİN BÖLÜTLEME VE**  
**SEYREK GÖSTERİM TEMELLİ SIKIŞTIRILMASI**

**ADEM ERTEM**

**Doç. Dr. M. Kemal GÜLLÜ**  
**Danışman, Kocaeli Üniversitesi**  
**Prof. Dr. Oğuzhan URHAN**  
**Jüri Üyesi, Kocaeli Üniversitesi**  
**Doç. Dr. Behçet Uğur TÖREYİN**  
**Jüri Üyesi, İstanbul Teknik Üniversitesi**

  
.....  
.....  
.....

**Tezin Savunulduğu Tarih: 04.02.2020**

## ÖNSÖZ VE TEŞEKKÜR

Bu tez kapsamında hiperspektral görüntülerin verimli bir şekilde sıkıştırılması için süperpiksel temelli özgün yöntemler geliştirilmiştir. Geliştirilen yöntemlerin veri sıkıştırma alanında çalışacak olan araştırmacılara faydalı olmasını dilerim.

Lisansüstü eğitimim sürecinde bilgi ve tecrübelerinden istifade ettiğim, akademik ilerlememde büyük katkıları olan değerli hocam ve tez danışmanım Doç. Dr. Mehmet Kemal Güllü'ye teşekkür ederim. Kıymetli fikirleriyle yol gösteren değerli hocam Prof. Dr. Oğuzhan Urhan'a teşekkür ederim. Ayrıca tez çalışmam boyunca en sıkışık zamanlarımda bilgi ve tecrübesiyle tezi tamamlamama büyük katkılar sağlayan Dr. Ali Can Karaca'ya teşekkür ederim.

Yüksek lisans eğitimim boyunca benden desteğini esirgemeyen ve hayatın bütün getirilerine karşı birlikte çaba harcadığımız sevgili eşim Tuba Ertem'e teşekkür ederim. Ayrıca hayatım boyunca desteklerini hissettiğim ve beni bugünlere getiren aileme sonsuz teşekkürlerimi sunarım.

Tez yazımında bilgi ve tecrübelerinden istifade ettiğim değerli arkadaşım Yasin Cuşkun'a teşekkür ederim.

Şubat – 2020

Adem ERTEM

## İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR .....	i
İÇİNDEKİLER .....	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ .....	vi
SİMGELER VE KISALTMALAR DİZİNİ .....	vii
ÖZET.....	viii
GİRİŞ .....	1
1. HİPERSPEKTRAL GÖRÜNTÜLEME.....	3
1.1. Testlerde Kullanılan Veri Kümeleri .....	7
1.2. Hesaplamada Kullanılan Metrikler .....	7
2. HİPERSPEKTRAL GÖRÜNTÜLERİN SIKIŞTIRILMASI.....	9
2.1. Giriş .....	9
2.1.1. Zamansal artıklık .....	9
2.1.2. Uzamsal artıklık .....	10
2.1.3. Spektral artıklık .....	10
2.1.4. İstatistiksel artıklık .....	11
2.2. Kayıpsız Sıkıştırma .....	11
2.3. Kayıplı Sıkıştırma.....	12
2.3.1. Dönüşüm temelli sıkıştırma .....	13
2.3.2. Sözlük öğrenme temelli sıkıştırma .....	14
2.3.3. Bölütleme temelli sıkıştırma .....	16
3. HİPERSPEKTRAL GÖRÜNTÜLERİN SÜPERPİKSEL VE SÖZLÜK ÖĞRENMESİ TEMELLİ SIKIŞTIRILMASI (SSASR) .....	19
3.1. Süperpiksel Bölütleme .....	20
3.2. Sözlük Öğrenmesi .....	23
3.3. Seyrek Katsayıların Belirlenmesi .....	25
3.4. Nicemleme, Entropi Kodlama ve Kod Çözme .....	29
4. SÜPERPİKSEL VE SEYREK GÖSTERİM TEMELLİ SIKIŞTIRMA İÇİN ÖNERİLEN YÖNTEMLER .....	30
4.1. Modifiye Edilmiş Sözlük ve Seyrek Gösterimle Süperpiksel Temelli Sıkıştırma Yöntemi (MSSASR) .....	31
4.1.1. Süperpiksel oluşturma .....	32
4.1.2. Sözlük oluşturma .....	33
4.1.3. Seyrek katsayıların belirlenmesi .....	37
4.1.4. Seyrek katsayıları düzenleme işlemi .....	39
4.1.5. Nicemleme, entropi kodlama ve kod çözme .....	41
4.1.6. Deneysel çalışmalar.....	41
4.2. Hata düzeltmeli süperpiksel temelli seyrek gösterimle sıkıştırma yöntemi (ECSSASR).....	49
4.2.1. Deneysel çalışmalar.....	52
4.3. Modifiye Edilmiş Sözlük Ve Seyrek Gösterimle Süperpiksel Temelli Anomali Düzeltmeli Sıkıştırma Yöntemi (MSSASR-RX) .....	55

4.3.1. Süperpiksel haritasının bulunması ve vektörleştirme işlemi.....	57
4.3.2. Sözlük öğrenmesi işlemi .....	58
4.3.3. Seyrek katsayıların belirlenmesi .....	60
4.3.4. Nicemleme entropi kodlama ve kod çözme .....	60
4.3.5. Sıkıştırma performansının güncel yöntemlerle karşılaştırılması.....	61
5. SONUÇLAR VE ÖNERİLER .....	68
KAYNAKLAR .....	70
KİŞİSEL YAYIN VE ESERLER .....	73
ÖZGEÇMİŞ .....	74



## ŞEKİLLER DİZİNİ

Şekil 1.1.	Hava platformundan alınan görüntü .....	3
Şekil 1.2.	Pasif ve aktif sistemlerin çalışma prensibi.....	4
Şekil 1.3.	Hiperspektral görüntü eksenleri.....	5
Şekil 1.4.	Toprak, dere ve bitkiye ait spektral imzalar.....	6
Şekil 1.5.	Kullanılan dört veri setinin yapay renklendirilmiş görselleri .....	7
Şekil 2.1.	Uzamsal artıklığı ifade için seçilen bölge ve piksel değerleri .....	10
Şekil 2.2.	İstatistiksel artıklığı ifade için hesaplama örneği .....	11
Şekil 2.3.	Tahmin temelli sıkıştırma işlem adımları .....	12
Şekil 2.4.	Dönüşüm işlem basamakları .....	13
Şekil 2.5.	Sözlük öğrenmesi temelli sıkıştırma yapısı .....	15
Şekil 2.6.	SLIC süperpiksel bölütleme yöntemi .....	16
Şekil 2.7.	SSASR yönteminin sıkıştırma başarımı (a) Indian Pines.(b) WashingtonDC. (c) Moffett. (d) Jasper Ridge için verilmiştir .....	17
Şekil 3.1.	SSASR yönteminin işlem aşamaları .....	19
Şekil 3.2.	Moffett görüntüsünde 100 ve 1000 adet süperpiksel için süperpiksel haritası.....	23
Şekil 3.3.	Sözlük oluşturmada süperpiksellerin gruplanması ve PCA katsayılarının bulunması .....	23
Şekil 3.4.	Sözlük oluşturmada alt sözlüklerin bulunması ve sözlüğün son hali.....	24
Şekil 3.5.	(a) Sözlük atomlarının 0,1 bit için kullanım durumu (b) Sözlük atomlarının 1 bit için kullanım durumu.....	25
Şekil 4.1.	Yöntemin genel işleyişi.....	31
Şekil 4.2.	Vektörleştirme işlemi .....	32
Şekil 4.3.	Süperpikselden örnek seçimi ve PCA katsayı matrisinin oluşumu .....	33
Şekil 4.4.	PCA sonucu elde edilen katsayılar yardımıyla sözlüğün oluşturulması .....	34
Şekil 4.5.	(a) Çoklu ve tekli PCA için sözlük oluşturma (b) Çoklu ve tekli PCA'de kullanılan atomların histogramı. ....	35
Şekil 4.6.	(a) Tek bir süperpiksel için seyrek katsayı haritası SSASR (b) Bütün veri için sıralanmış seyrek katsayı haritası SSASR.....	36
Şekil 4.7.	(a) Tek bir süperpiksel için seyrek katsayı dağılım haritası MSSASR (b) Bütün veri için sıralanmış seyrek katsayı dağılım haritası MSSASR .....	37
Şekil 4.8.	Seyrek katsayı matrisi oluşturulurken izlenen yol .....	40
Şekil 4.9.	Farklı süperpiksel sayısı ve örnek seçimi için SNR-Bit grafikleri .....	42
Şekil 4.10.	Tekli PCA ve DPCM işleminin yöntem katkısı.....	43
Şekil 4.11.	Washington DC için metrik sonuçları .....	44
Şekil 4.12.	Indian Pines için metrik sonuçları .....	45
Şekil 4.13.	Jasper Ridge için metrik sonuçları.....	46
Şekil 4.14.	Moffett Field için metrik sonuçları.....	47

Şekil 4.15. (a) Jasper Ridge verisi için ortalama SNR grafiği (b) Moffett Field verisi için ortalama SNR grafiği (c) Moffett için anomali haritası .....	48
Şekil 4.16. Yöntemin genel işleyişi .....	49
Şekil 4.17. Kullanılan veri (a) Washington DC ve (b) hata haritası .....	51
Şekil 4.18. Hatalı pikseller için oluşturulan sözlük ve katsayılar .....	51
Şekil 4.19. Sıkıştırma aşamaları.....	52
Şekil 4.20. (a) SSASR SNR dağılımı. (b) ECSASR SNR dağılım .....	53
Şekil 4.21. Yöntemlere ait metrik sonuçları.....	54
Şekil 4.22. MSSASR-RX yönteminin işlem aşamaları.....	56
Şekil 4.23. Anomali olan ve olmayan pikseller için vektörel dönüşüm.....	57
Şekil 4.24. MSSASR-RX için sözlük öğrenme işlem basamakları .....	58
Şekil 4.25. MSSASR-RX için sözlük öğrenme işlem basamakları .....	59
Şekil 4.26. Moffett Field için metrik sonuçları .....	62
Şekil 4.27. Indian Pines için metrik sonuçları .....	63
Şekil 4.28. Jasper Ridge için metrik sonuçları .....	64
Şekil 4.29. Washington DC için metrik sonuçları .....	65
Şekil 4.30. 150 anomali piksel için dört farklı veride ortalama SNR değeri .....	66
Şekil 4.31. Moffett verisi için (a) SNR dağılım haritası (b) hata haritası verilmiştir (c) ise anomali dağılım haritası verilmektedir.....	67

## TABLÖLAR DİZİNİ

Tablo 3.1. Süperpiksel oluřturma.....	22
Tablo 3.2. Adaptif seyrek kodlama .....	28
Tablo 3.3. Kod çözme .....	29
Tablo 4.1. Uyarlamalı seyrek kodlama .....	39





## SİMGELER VE KISALTMALAR DİZİNİ

### Kısaltmalar

AVIRIS	: Airborne Visible / Infrared Imaging Spectrometter (Hava Platformunda Görünür / Kızılötesi Spektrometresi)
BP	: Basis Pursuit (Temel Takip Algoritması)
BCO	: Bulanık C-Ortalamlar
DWT	: Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)
DPCM	: Differential Pulse Code Modulation (Ayrımsal Darbe Kod Kiplenimi)
ECSSASR	: Error Correcting Spectral Spatial Adaptive Sparse Representation (Hata Düzeltmeli Spektral Uzamsal Uyarlamalı Seyrek Tanımlama)
GKK	: Bulanık Gustafson Kessel
JPEG	: Joint Photographic Experts Group (Birleşik Fotoğraf Uzmanları Gurubu)
K-SVD	: K-Singular Value Decomposition (K adet – Tekil Değer Ayırıştırma)
LiDAR	: Light Detection and Ranging (Işık Tespiti ve Ölçümü)
MAD	: Maximal Absolute Difference (Maksimum Mutlak Fark)
MAE	: Mean Absolute Error (Ortalama Mutlak Hata)
MSA	: Maximum Spectral Angle (Maksimum Spektral Açığı)
MSC	: Mean Spectral Correlation (Ortalama Spektral İlinti)
MSE	: Mean Square Error (Ortalama Karesel Hata)
MSSASR	: Modified Spectral Spatial Adaptive Sparse Representation (Değiştirilmiş Spektral Uzamsal Uyarlamalı Seyrek Tanımlama)
OMP	: Orthogonal Matching Pursuit (Dikgen Eşleşen Arama )
PCA	: Principal Components Analysis (Temel Bileşenler Analizi)
PSNR	: Peak of Signal to Noise Ratio (Tepe Sinyal Gürültü Oranı)
RADAR	: Radio Detection And Ranging (Radyo Sinyali Algılama ve Mesafe Tayini)
ROI	: Region Of Interest (İlgi Alanı)
SLIC	: Simple Linear Iterative Clustering (Basit Liner İteratif Kümeleme)
SNR	: Signal to Noise Ratio(Sinyal Gürültü Oranı)
SOMP	: Simultaneous Orthogonal Matching Pursuits (Eş Zamanlı Dikgen Eşleşen Arama)
SSASR	: Spectral Spatial Adaptive Sparse Representation (Spektral Uzamsal Uyarlamalı Seyrek Tanımlama)

# HİPERSPEKTRAL GÖRÜNTÜLERİN BÖLÜTLEME VE SEYREK GÖSTERİM TEMELLİ SIKIŞTIRILMASI

## ÖZET

Hiperspektral görüntüler, yüksek miktarda veri içermektedir. Verinin iletilmesi ve saklanması maliyetli yapılar gerektirdiği için yüksek veri miktarları azaltılmak istenmektedir. Bu sebeple, iletimde bant genişliğini düşürmek ve depolamada daha az bellek alanı kullanmak için veri sıkıştırılmaya ihtiyaç duyulmaktadır.

Bu çalışma kapsamında, hiperspektral verileri daha iyi sıkıştırabilmek için özgün yöntemler geliştirilmiştir. Geliştirilen yöntemler süperpiksel bölütleme temelli olup sözlük öğrenmesi ve seyrek gösterim işlemleri barındırmaktadır. Önerilen yöntemler veriyi ilk olarak süperpikselle ayırmaktadır. Daha sonra sözlük oluşturmada kullanmak için her bir süperpikselden belirli sayıda rastgele spektral imza alınmaktadır. Alınan bu imzalar PCA yardımıyla sözlük için atomlara dönüştürülmektedir. Oluşturulan sözlük yardımıyla her bir süperpiksel için seyrek katsayılar hesaplanmaktadır. Son olarak, hesaplanan seyrek katsayılar ve sözlük entropi kodlama yardımıyla sıkıştırılıp iletilmektedir.

Geliştirilen yöntemler yüksek veri sıkıştırma oranlarına çıkılmasına katkı sağlamaktadır. Ayrıca geliştirilen yöntemler, anomali olan piksellerin korunmasında verimli bir şekilde çalışmaktadır. Son olarak önerilen yöntemlerin, literatürdeki güncel kayıplı sıkıştırma yöntemleriyle performansı birçok metrikle kıyaslanmaktadır.

**Anahtar Kelimeler:** Hiperspektral Görüntü Sıkıştırma, Seyrek Gösterim, Sözlük Öğrenmesi, Süperpiksel Bölütleme.

## **SEGMENTATION AND SPARSE REPRESENTATION BASED COMPRESSION OF HYPERSPECTRAL IMAGES**

### **ABSTRACT**

Hyperspectral images contain large amounts of data. Since data transmission and storage require costly structures, high amounts of data are desired to be reduced. Therefore, data compression is needed to reduce bandwidth in transmission and to use less memory space in storage.

In this study, specific methods have been developed to better compress the hyperspectral data. The methods developed are based on superpixel segmentation and include dictionary learning and sparse representation. The proposed methods first divide the data into superpixels. Then a certain number of random spectral signatures are obtained from each superpixel for use in dictionary creation. These signatures are converted into atoms for a dictionary using PCA. With the help of the generated dictionary, sparse coefficients are calculated for each superpixel. Finally, the calculated sparse coefficients and the dictionary are compressed and transmitted with the help of entropy coding.

The developed methods contribute to high data compression rates. In addition, the developed methods work efficiently in the protection of anomalous pixels. Finally, the performance of the proposed methods with the current lossy compression methods in the literature is compared with many metrics.

**Keywords:** Hyperspectral Image Compression, Sparse Representation, Dictionary Learning, Superpixel Segmentation.

## **GİRİŞ**

Hiperspektral görüntü yüzlerce spektral banda sahiptir. Her bir bant, görüntüsü alınan bölge hakkında anlamlı bilgiler barındırmaktadır. İlgili bölge hakkında elde edilen anlamlı bilgiler yardımıyla, birçok uygulama alanı ortaya çıkmıştır. Birçok uygulama alanının olması hiperspektral görüntülemeyi gün geçtikçe cazibedar hale getirmektedir. Özellikle askeri uygulamalarda, bitki örtüsü haritasının çıkarımında, su kaynakları ve tarımsal ürün türünün tespit edilmesi gibi uygulamalarda sıkça kullanılmaktadır.

Hiperspektral veriler yüzlerce bant içerdiği için veri boyutu büyük olmaktadır. Bu sebeple iletilmesinde ve depolamasında sorunlar yaşanmaktadır. Bu sorunları ortadan kaldırmak için sıkıştırma teknolojisi ortaya çıkmıştır. Çeşitli sıkıştırma algoritmaları yardımıyla, veri boyutu küçültülerek, iletimin kolaylaştırılması sağlanmak istenmektedir. Bu kapsamda verileri sıkıştırabilmek için kayıplı, kayıpsız ve kayıpsıza yakın sıkıştırma teknikleri oluşturulmuştur.

Tez çalışması kapsamında, kayıplı veri sıkıştırma algoritmaları incelenmektedir. Kayıplı veri sıkıştırma algoritmaları, yüksek veri sıkıştırma oranlarına sahip olduğu için tercih edilmektedir. Süperpiksel bölütleme temelli kayıplı sıkıştırma yöntemi bu kapsamda bölümler halinde incelenmektedir.

Bölüm 1’de uzaktan algılama sistemleri hakkında çeşitli bilgiler verilmektedir. Örneğin hangi platformlar yardımıyla elde edildikleri ve alınan görüntülerin farkları ele alınmaktadır. Hiperspektral görüntüleme hakkında bilgi verildikten sonra, testlerde kullanılan veri kümeleri açıklanmaktadır.

Bölüm 2’de literatürde kullanılan sıkıştırma teknikleri hakkında genel bilgi verildikten sonra kayıplı sıkıştırma teknikleri detaylandırılmaktadır.

Bölüm 3’te, literatürde bulunan kayıplı sıkıştırma yöntemlerinden süperpiksel ve sözlük öğrenmesi temelli hiperspektral görüntülerin sıkıştırılması yöntemi detaylı bir

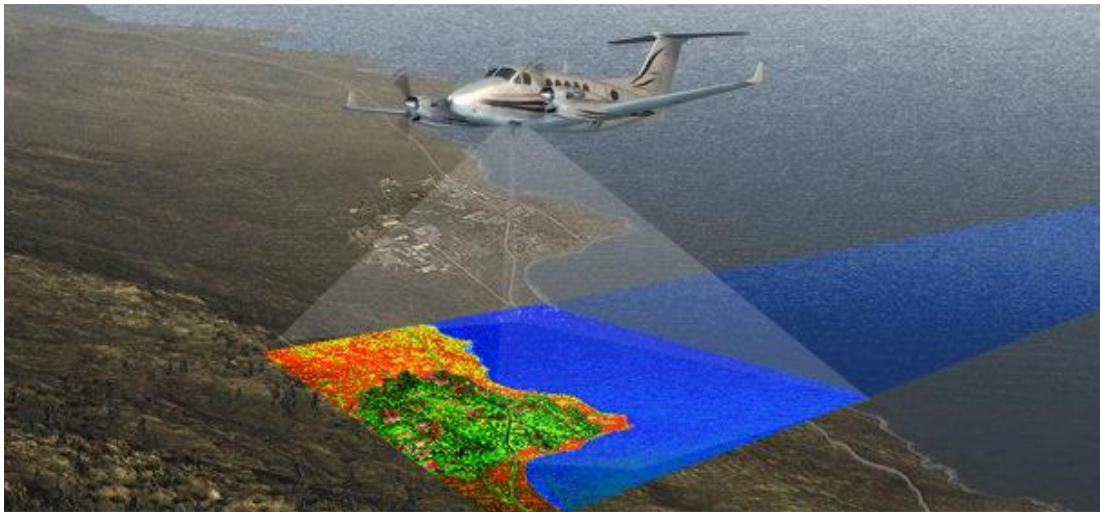
şekilde anlatılmaktadır. Bu yöntemin işlem adımlarındaki algoritmalar verilerek, çalışması ve özellikleri detaylı anlatılmaktadır.

Bölüm 4'te, tez çalışması kapsamında geliştirilen yöntemlerin çalışma şekilleri sırası ile anlatılmaktadır. Geliştirilen yöntemler MSSASR, ECSSASR ve MSSASR-RX olmak üzere üç ana başlık altında toplanmaktadır. ECSSASR yöntemi SSASR yönteminin eksiklerinden biri olan bazı pikselleri aşırı bozarak ifade edememe sebebini, yüksek hatalı ifade edilen pikseller için yeni sözlük tasarlayarak ortadan kaldırmaktadır. MSSASR yönteminde, tüm süperpikselleri en iyi ifade edecek yeni bir sözlük tasarımı yapılmaktadır. Ayrıca iletim öncesi seyrek katsayıları süperpiksel haritasının alıcıya iletim ihtiyacını ortadan kaldıracak şekilde sıralamaktadır. Son olarak tasarlanan yeni sözlük sayesinde seyrek katsayıların enerjisinin büyük kısmı ilk satırlarda toplanmaktadır bu nedenle yüksek genlik değerlerine sahip seyrek katsayılara DPCM uygulanmaktadır. MSSASR-RX yönteminde ise hem MSSASR yönteminin sözlük tasarımından faydalanılmakta hem de anomali pikseller tespit edilerek onlar için ayrı bir sözlük tasarımı önerilerek anomali pikseller daha verimli sıkıştırılmaktadır. Ayrıca diğer sıkıştırma yöntemleriyle faklarına değinilerek avantajları göz önüne çıkarılmaktadır. Deneysel sonuçlar paylaşılarak, geliştirilen yöntemlerin katkısı grafikler halinde ortaya konmuştur.

## 1. HİPERSPEKTRAL GÖRÜNTÜLEME

Uzaktan algılama, atmosfer ve uzaydaki platformlara yerleştirilmiş ölçüm aletleri yardımıyla, yeryüzündeki nesnelere hakkında bilgi toplama ve bunları analiz etme tekniği, veya herhangi bir nesneye fiziksel temasta bulunmadan belirli bir uzaklıktan nesnelere hakkında bilgi toplama bilim ve sanatı olarak ifade edilir [1]. Uzaktan algılama teknolojileri kullanılarak, su kalitesi tespiti, sulak alan haritalama, termal anomaliler, jeoloji, madencilik, yağ ve gaz araştırma, savunma ve hedef bulma, çevre ile ilgili konularda araştırmalar yapılmaktadır. Yapılan bu araştırmalara termal anomaliler konu başlığı altında yanmış kömür damarı haritalama, su kalitesi konu başlığı altında kirlilik varlığı ve yayılımı, tarımda ise hastalık ve stres haritalama gibi konular örnek verilmektedir.

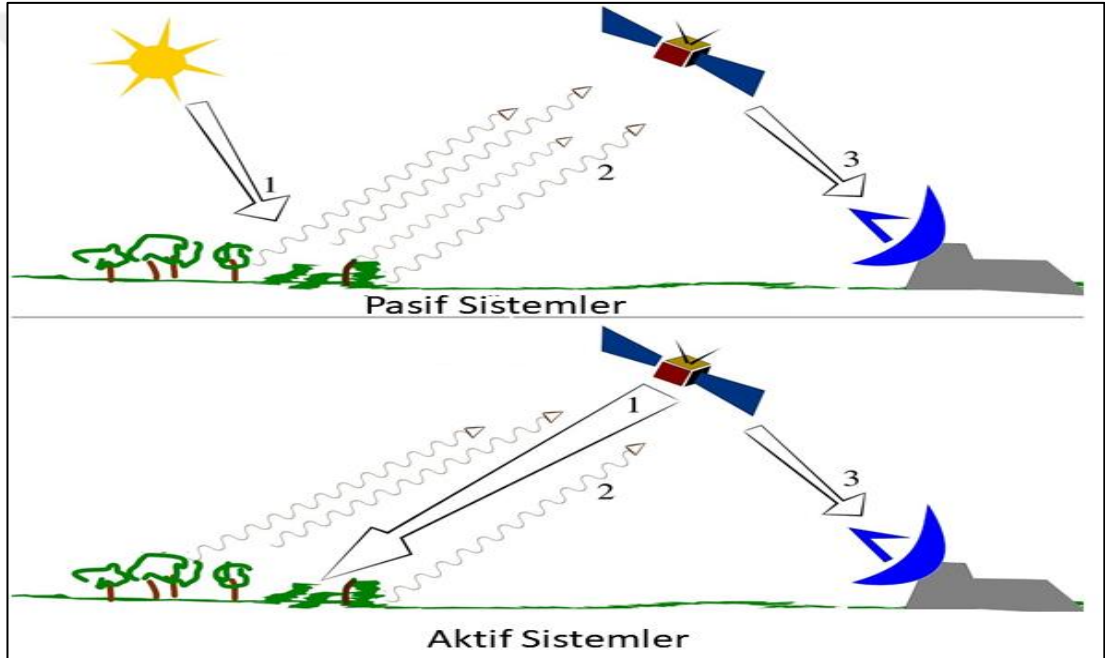
Uzaktan algılama yapan sistemler kullandıkları platformlara göre yer platformları, hava platformları, uydu platformları olmak üzere üçe ayrılmaktadır. Yer platformları, çok fazla ayrıntıya ihtiyaç duyulan sistemlerde kullanılır. Uzamsal çözünürlükleri yüksek olmakla beraber kapsadıkları alan düşüktür. Hava platformlarından elde edilen görüntüler daha çok uçak, insansız hava aracı, helikopter gibi sistemlere entegre edilerek elde edilen verilerden oluşmaktadır.



Şekil 1.1. Hava platformundan alınan görüntü [2]

Şekil 1.1’de hava platformundan alınan veri gösterilmektedir. Hava platformlarından alınan verilerin uzamsal çözünürlükleri yer platformlarına göre düşük olmakla beraber, kapsadıkları alan yer platformlarına göre daha yüksektir. Uydu platformları ise uydulara yerleştirilen sensörler yardımıyla alınan verileri içermektedir. Bu sistemler yardımıyla elde edilen verilerin, uzamsal çözünürlükleri düşük olmakla beraber kapsadıkları alan fazladır.

Uzaktan algılama sistemleri ölçüm biçimlerine göre pasif algılayıcılar ve aktif algılayıcılar olmak üzere ikiye ayrılmaktadır. Bu sistemlerin genel işleyiş şeması Şekil 1.2’de verilmektedir.

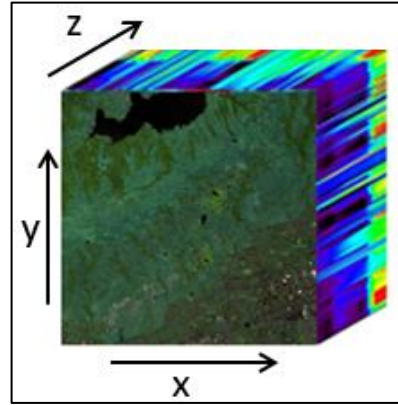


Şekil 1.2. Pasif ve aktif sistemlerin çalışma prensibi [4]

Pasif sistemler ısı ve ışık kaynağına ihtiyaç duymadan güneş enerjisini kullanarak ölçüm yapabilen sistemlerdir. Termal kameralar buna örnek olarak verilebilir. Aktif sistemlerin kendilerine özgü kaynakları vardır. Yayıdıkları elektromanyetik enerji cisim yüzeylerinden yansıyarak saçılım oluşturmaktadır. Bu sistemler geriye yansıyan sinyalleri yakalayıp algılama yaparlar. Uzaktan algılama sistemlerinde sıkça kullanılan algılayıcılar altı ana başlık altında toplanabilmektedir. Bunlar pankromatik, multispektral, hiperspektral, ultraspektral, LiDAR ve RADAR sistemleridir. LiDAR üç boyutlu lazer tarama yöntemiyle çalışmaktadır. İlk olarak

havadan deniz altı araçlarının bulunması için geliştirilmiş aktif bir sistemdir. RADAR sistemleri ise cisimlerin konumunu büyüklük ve hız bilgisini çıkarmak için radyo dalgalarını kullanan aktif bir sistemdir. RADAR'ın gönderdiği elektromanyetik sinyal cisim yüzeyine çarptıktan sonra geri yansımaktadır. Yansıyan bu sinyallerin açısı şiddeti ve geri gelme süresine bağlı olarak cisim hakkında bilgiler elde edilir. Nesnelerin ışığı farklı dalga boylarında farklı yansıtma prensibine dayanarak multispektral, hiperspektral ve ultraspektral görüntüleme teknikleri ortaya çıkmıştır. Bu görüntüleme sistemleri arasındaki temel farklar içerdikleri bant sayısıdır. Multispektral veriler ortalama on bant içerirken hiperspektral veriler yüzlerce ultraspektral veriler ise binlerce bant içermektedir. Pankromatik veriler ise düşük spektral veri içermesinin yanında yüksek uzamsal bilgi içermektedir.

Hiperspektral görüntüler, yüzeyde bulunan cisimlerden yansıyan enerjinin, art arda yerleştirilmiş çok sayıda dalga boyu bandında ölçülmesi ile elde edilirler. Hiperspektral görüntüler Şekil 1.3'te gösterildiği gibi x ve y ekseninde uzamsal bilgiyi barındırırken z ekseninde ise spektral bilgiyi barındırmaktadır. Uzamsal eksenlerdeki her bir x ve ye koordinatında, ilgili bölgenin ilgili banttaki piksel değeri bulunmaktadır.

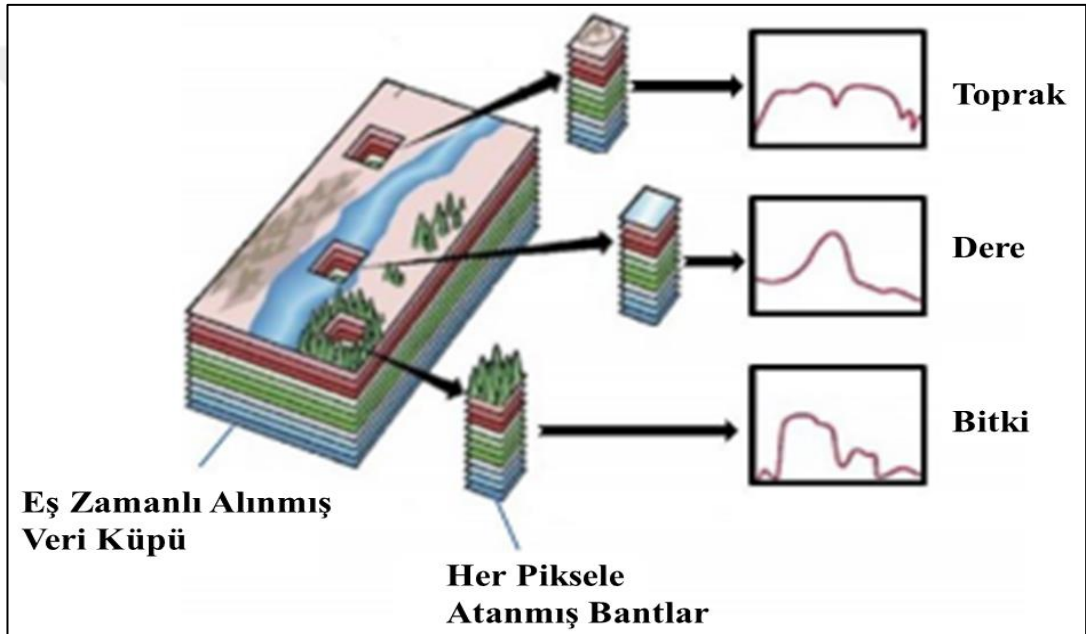


Şekil 1.3. Hiperspektral görüntü eksenleri

Aynı piksel koordinatına sahip art arda gelen bantlardaki piksel değerlerinin birleşmesiyle ilgili bölgenin spektral imzası oluşmaktadır. Spektral imza, cisim yüzeyinden yansıyan ışının farklı dalga boylarındaki enerjisinin ölçülmesi ile elde edilir. Cisimlerin yansımaları farklı dalga boylarında farklılık göstermektedir. Bir



kısım maddeler bazı dalga boylarında yüksek yansımaya değerlerine sahipken bazı maddeler aynı dalga boyunda emilmeye uğrayarak daha az yansımaya yapmaktadır. Cisimlerin farklı dalga boylarındaki farklı yansımalarından faydalanılarak her bir madde için spektral kimlikler oluşturulabilmektedir. Spektral imza farklılıklarını ifade için Şekil 1.4'te toprak su ve bitki için spektral imzalar gösterilmektedir. Normal şartlarda gözle ayrılması imkansız olan farklılıklar spektral imzalar sayesinde kolayca fark edilebilmektedir. Spektral imzalar yukarıda bahsettiğimiz özelliklerinden dolayı kamuflej tespiti [13], maden aramaları, bitki sağlığı, askeri hedef tespiti [19] gibi birçok uygulamada yaygın olarak kullanılmaktadır.

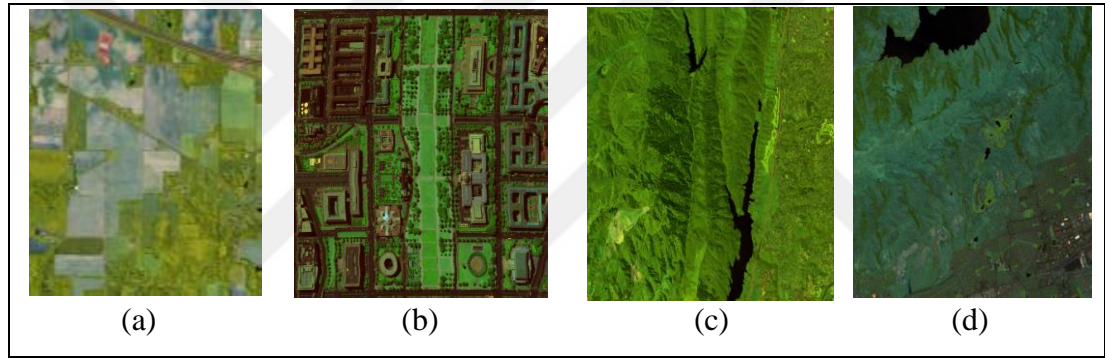


Şekil 1.4. Toprak, dere ve bitkiye ait spektral imzalar [4]

Hiperspektral görüntüler avantajlara sahip olduğu gibi dezavantajları da mevcuttur. Genellikle bu tür sistemlerin uzamsal çözünürlükleri düşük olduğu için uzamsal eksende her bir piksel içerisine farklı materyaller karışabilmektedir. Dolayısıyla spektral imzalar bu karışımdan dolayı bazen cisimlerin kimlikleri hakkında sağlıklı bilgi verememektedir. Bir başka sorun ise veri boyutundaki büyüklüktür. Hiperspektral verilerde yüzlerce spektral bant bulunmaktadır bu sebeple yüksek miktarda bilgi içermektedir. Hiperspektral görüntülerin içerdikleri yüksek miktarda veri, iletim problemini ortaya çıkarmıştır. Ortaya çıkan bu problemi kaldırmak için veri sıkıştırma algoritmaları önem kazanmıştır.

## 1.1. Testlerde Kullanılan Veri Kümeleri

Deneysel çalışmalar boyunca yöntemlerin performanslarının karşılaştırılması için Şekil 1.5(a)'da Indian Pines, Şekil 1.5(b)'de DC Mall, Şekil 1.5(d)'de Moffett Field ve Şekil 1.5(c)'de Jasper Ridge olmak üzere toplam dört farklı veri kullanılmıştır. Bu verilerden Indian Pines verisinden gürültülü bantlar atılmış ve  $145 \times 145 \times 200$  boyutlarındaki veri kullanılmıştır. Washington DC görüntüsü [5]'deki çalışmaya benzer olarak  $280 \times 307$  boyutlarında kırılarak kullanılmıştır. DC Mall görüntüsü toplam 191 spektral bant içermektedir. Jasper Ridge ve Moffett Field görüntüleri ise aynı boyutlara sahip olup  $512 \times 512 \times 224$  boyutlarındadır. Tüm veriler 16 bit derinliğine sahiptir. Şekil 1.5'te bütün veriler için yapay renklendirilmiş görseller verilmektedir.



Şekil 1.5. Kullanılan dört veri setinin yapay renklendirilmiş görselleri

## 1.2. Hesaplama Kullanılan Metrikler

Kayıplı sıkıştırmada önerilen yöntemin performansının literatürdeki yöntemlerle karşılaştırılması için Denklem 1.1'de SNR, Denklem 1.3'te ortalama spektral açı benzerliği (Mean Spectral Angle, MSA), Denklem 1.6'da ortalama spektral ilinti (Mean Spectral Correlation, MSC), Denklem 1.7'de Pik Sinyal Gürültü Oranı (Peak Signal to Noise Ratio, PSNR), Denklem 1.2'de Ortalama Mutlak Hata (Mean Absolute Error, MAE), Denklem 1.5'te Ortalama Karesel Hatanın (Mean Squared Error, MSE) gibi benzerlik ve uzaklık metrikleri kullanılmaktadır. Burada  $I$  ve  $I^R$ ,  $M \times N \times B$  boyutlu orijinal ve sıkıştırma sonrasında geri elde edilen görüntüyü,  $Corr$  ise ilinti katsayısını göstermektedir. SNR, MSC, PSNR benzerlik metriği, MSE, MSA ve MAE uzaklık metriğidir. Tabloda uzaklık metriği olarak paylaşılan

metrikler görüntüler arasındaki hatayı sorgulamakta ve küçük değerleri iyi olarak yorumlanmaktadır. Denklem 1.4'te spektral açı hesabı gösterilmektedir.

$$\text{SNR} = 10 \log_{10} \left( \frac{\sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^B I(m, n, p)^2}{\sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^B [I(m, n, p) - I^R(m, n, p)]^2} \right) \quad (1.1)$$

$$\text{MAE} = \frac{1}{M \times N \times B} \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^B |I(m, n, p) - I^R(m, n, p)| \quad (1.2)$$

$$\text{MSA} = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N \text{SA}_{m,n} \quad (1.3)$$

$$\text{SA}_{m,n} = \cos^{-1} \left( \frac{\langle I(m, n), I^R(m, n) \rangle}{\|I(m, n)\| \|I^R(m, n)\|} \right) \quad (1.4)$$

$$\text{MSE} = \frac{1}{M \times N \times B} \sum_{m=1}^M \sum_{n=1}^N \sum_{p=1}^B [I(m, n, p) - I^R(m, n, p)]^2 \quad (1.5)$$

$$\text{MSC} = \frac{1}{M \times N \times B} \sum_{m=1}^M \sum_{n=1}^N \text{Corr}(I(m, n), I^R(m, n)) \quad (1.6)$$

$$\text{PSNR} = 20 \log_{10} \left( \frac{2^{16} - 1}{\sqrt{\text{MSE}}} \right) \quad (1.7)$$

Bit oranı hesabında her bir banttaki piksel başına düşen bit oranı hesaplanmaktadır ve bpp olarak gösterilmektedir. Benzerlik metrikleri ise uzaklık metriklerinin tam tersine büyük değerler verdiğinde hiperspektral görüntünün iyi bir ölçüde geri elde edildiğini göstermektedir. SNR sinyal gücünün gürültü gücüne oranını ifade etmektedir. MSC spektral imzalar arası ilintiyi ifade etmektedir. MSA spektral imzalar arası açısal uzaklığı ifade etmektedir. MAE ortalama mutlak hatayı ifade etmektedir. MAE spektral ilinti azaldıkça artmaktadır. MSE ise ortalama karesel hatadır.

## 2. HİPERSPEKTRAL GÖRÜNTÜLERİN SIKIŞTIRILMASI

### 2.1. Giriş

İnsanlık tarihi ilerledikçe, toplumların sahip olduğu bilgi birikimi geçmişten on dokuzuncu yüzyıla kadar yazılı ve sözlü olarak aktarılarak gelmiştir. Teknolojinin gelişmesiyle bu yöntemlere ses kaydı, resim, video gibi bilgi aktarım yöntemleri de eklenmiştir. İnternetin hayatımıza girmesi ile birlikte bu işlem başka bir boyuta taşınarak, kişilerin bilgiye erişim ve yeni bilgi aktarma kapasitesi giderek artmıştır. Bu artan bilgi kapasitesini verimli bir şekilde yönetebilmek için verimli bir iletim ve depolama ihtiyacı ortaya çıkmaktadır. Bu ihtiyaç sonucu aynı veriyi daha küçük boyutta iletmek ve depolamak için sıkıştırma algoritmaları ortaya çıkmıştır. Veri sıkıştırma işleminin asıl amacı, verinin iletilmesi için gerekli bant genişliğini düşürmek ve depolama için gerekli alanı en aza indirmektir. Teknolojik gelişmeler veri sıkıştırma ihtiyacını kaçınılmaz hale getirmiştir. Buna bağlı olarak ta, veri sıkıştırma algoritmaları giderek popüler hale gelmektedir. Veri sıkıştırma algoritmaları temel olarak kayıplı sıkıştırma ve kayıpsız sıkıştırma olmak üzere ikiye ayrılmaktadır. Kayıpsız sıkıştırmada amaç, geri elde edilen verinin orijinalle aynı kalmasını sağlarken sıkıştırılmış veri boyutunu en aza indirebilmektir. Kayıplı sıkıştırma işlemi sonucunda orijinal veride kayıplar oluşmaktadır bu kayıplar bozunum olarak ifade edilmektedir. Kayıplı sıkıştırmanın amacı ise bozunumu en aza indirerek, en yüksek sıkıştırma oranlarına çıkabilmektir. Veri sıkıştırma işlemi sırasında sıkıştırma algoritmaları verideki zamansal, uzamsal, istatistiksel ve spektral artıklıkları dikkate almaktadır. Verideki artıklıklar ne kadar giderilirse sıkıştırma o oranda artmaktadır.

#### 2.1.1. Zamansal artıklık

Video art arda gelen görüntü çerçevelerinden oluşmaktadır. Görüntü kaydı yapan cihaz, sabit bir sisteme entegre edilmişse, birbirini takip eden çerçeveler arasında değişim çok az olmaktadır. Çerçeveler arasındaki değişimin az olması yüksek ilinti

var olduğu anlamına gelmektedir. Birbirini takip eden çerçeveler arasında ilintinin yüksek olması zamansal artıklığın az olduğunu göstermektedir.

### 2.1.2. Uzamsal artıklık

Komşu olan pikseller birbirine genlik olarak benzeme eğilimindedirler. Bundan dolayı komşu plan piksellerin ilintileri oldukça yüksektir. Komşu olan piksellerin ilintileri yüksek olduğu gibi değişimin az olduğu görüntülerde bölgesel ilintiler yüksek olmaktadır.

Bu ilinti çeşitli yöntemlerle giderilerek veri sıkıştırma işlemleri gerçekleştirilmektedir. Şekil 2.1'de mavi ile seçilen bölgenin piksel değerleri görülmektedir. Görüldüğü gibi komşu olan piksellerin tamamı 254 değerine sahiptir.



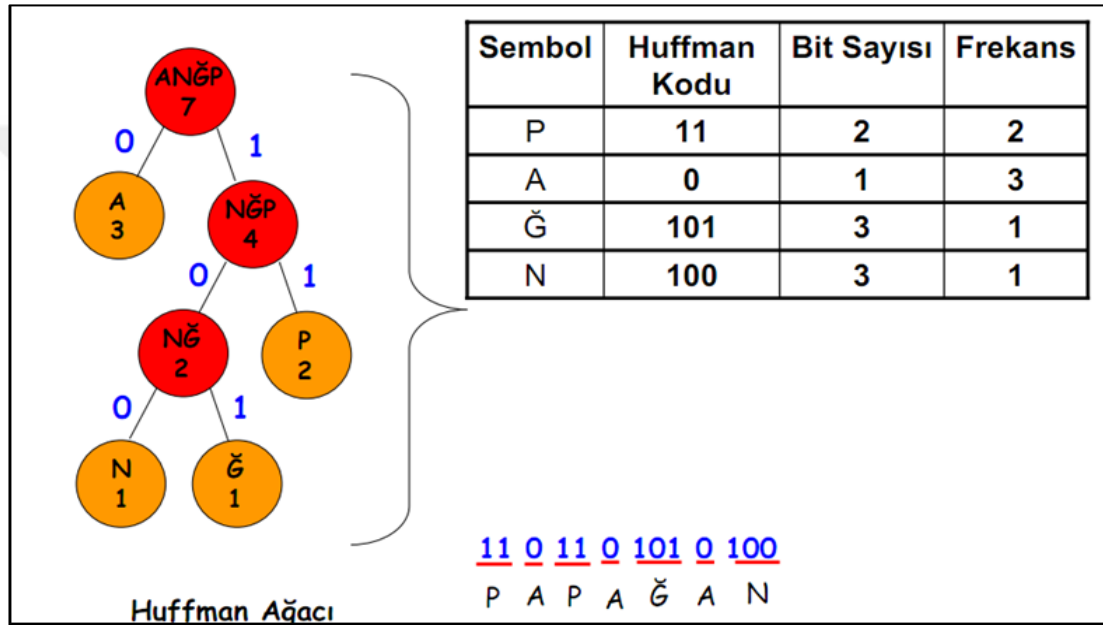
Şekil 2.1. Uzamsal artıklığı ifade için seçilen bölge ve piksel değerleri

### 2.1.3. Spektral artıklık

Bant sayısının fazla olduğu renkli, multispektral ve hiperspektral görüntülerde bantlar arasındaki ilintiyi ifade etmektedir.

#### 2.1.4. İstatistiksel artıklık

Birbirini tekrar eden semboller istatistiksel artıklık oluşturmaktadır. Bu artıklığı ortadan kaldırmak için aritmetik kodlama Huffman kodlama gibi yöntemler kullanılmaktadır. Bu sistemlerin çalışma şekli, veri içerisinde yüksek tekrar oranına sahip sembolleri kısa uzunluklu kodlarla, düşük tekrar oranına sahip sembolleri yüksek uzunluğa sahip kodlarla ifade etme işlemidir. İstatistiksel artıklık genel olarak diğer artıklıklar giderildikten sonra veriye uygulanan bir işlemdir [8].



Şekil 2.2. İstatistiksel artıklığı ifade için hesaplama örneği [3]

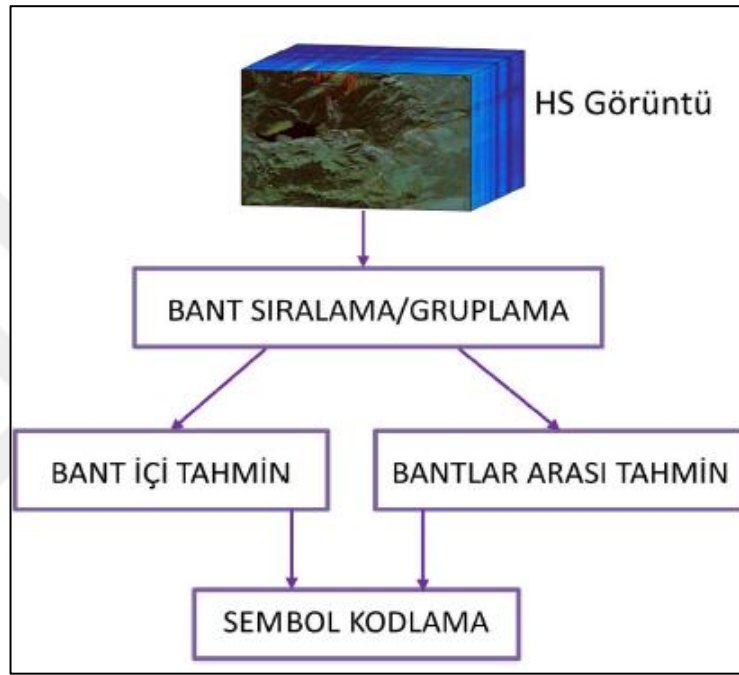
Şekil 2.2’de Huffman kodlama örneği gösterilmektedir papağan kelimesindeki her bir harfi 8 bit ile kodlayıp göndermek yerine sadece 13 bit ile bütün kelimeyi ifade etmektedir.

#### 2.2. Kayıpsız Sıkıştırma

Kayıpsız sıkıştırmada, geri elde edilen veri orijinal veriyle aynı kalmaktadır. Kayıpsız sıkıştırma işlemleri, kayıplı sıkıştırma işlemlerine kıyasla yüksek sıkıştırma oranlarına çıkamamaktadır. Hiperspektral verilerde kayıpsız sıkıştırma işlemi gerçekleştirilirken, verinin uzamsal ve spektral artıklığından faydalanılmaktadır. Hiperspektral görüntülerin, uzamsal ve spektral eksenindeki birbirine yakın pikseller benzerlik göstermektedir. Bu benzerlikten faydalanılarak tahmin temelli kayıpsız

sıkıştırma algoritmaları geliştirilmektedir. Kayıpsız sıkıştırma yöntemlerinin temelini tahmin temelli yöntemler oluşturmaktadır [8].

Kayıpsız sıkıştırmada, tahmin temelli yöntemler uygulandıktan sonra sembolik artıklığı ortadan kaldırmak için Huffman ve aritmetik kodlama ile sembol kodlama yapılmaktadır. Şekil 2.3'te tahmin temelli sıkıştırmanın işlem adımları gösterilmektedir.



Şekil 2.3. Tahmin temelli sıkıştırma işlem adımları [8]

### 2.3. Kayıplı Sıkıştırma

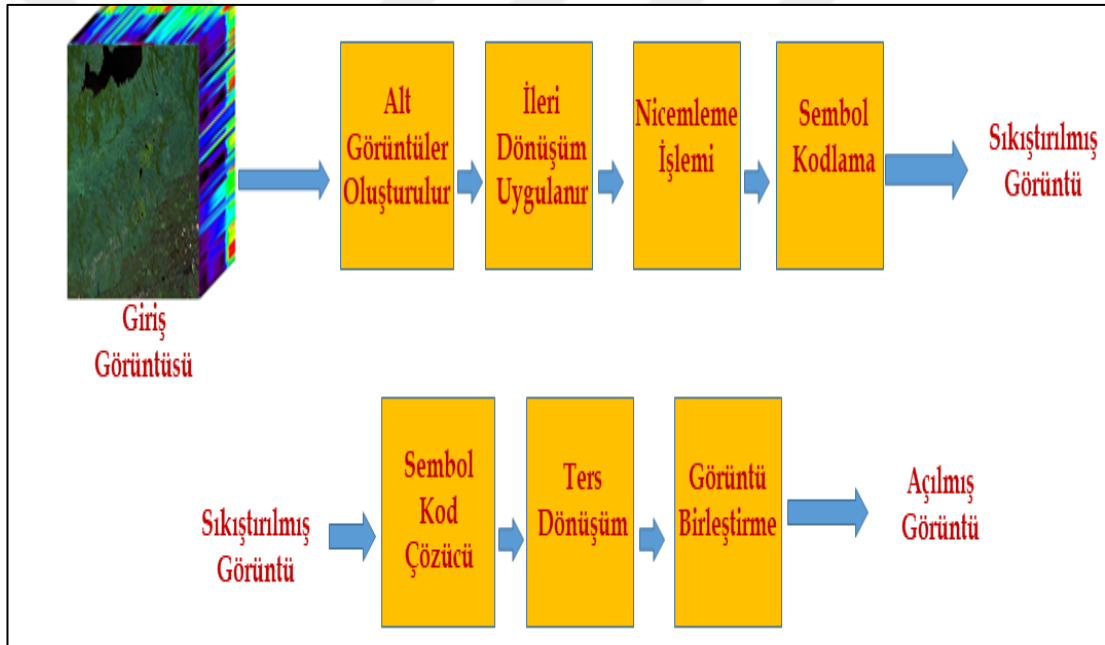
Görüntüler, kullanıcının ilgilendiği veriler ve ilgilenmediği verilerden oluşmaktadır. Kullanıcının ilgilendiği noktalar yeteri kadar koruna bildiği müddetçe diğer kısımlardaki veri kayıpları önemli olmamaktadır. Veri kaybının önemsiz olduğu sistemlerde kayıplı sıkıştırma önerilmektedir. Kayıplı sıkıştırma yönteminde amaç yüksek sıkıştırma oranlarında minimum bozunumu elde etmektir. Kayıplı sıkıştırma algoritmaları sayesinde yüksek sıkıştırma oranlarına çıkılabilmektedir. Veri iletim maliyetlerinin yüksek öneme sahip olduğu günümüzde, kayıplı veri sıkıştırma yöntemleri her geçen gün önemini artırmaktadır. Kayıplı sıkıştırmada pek çok

yöntem kullanılmaktadır. Bu yöntemler bölütleme temelli, dönüşüm temelli, sözlük öğrenmesi temelli yöntemler ön plana çıkmaktadır.

### 2.3.1. Dönüşüm temelli sıkıştırma

Dönüşüm temelli yöntemler veriyi bazı dönüşümlerden geçirerek uzamsal ve spektral artıklığı kaldırmaktadır. Dönüşüm temelli yöntemler JPEG, JPEG2000, MPEG-2 ve MPEG-4 gibi yöntemlerde kullanılmaktadır. Dönüşüm işleminde birçok dönüşüm metodu önerilmektedir. Dönüşüm işlemleri blok temelli ya da görüntü temelli olabilmektedir.

Bloğa dayalı yöntemlerin işlem basamakları Şekil 2.4'te verilmektedir.



Şekil 2.4. Dönüşüm işlem basamakları

Bloğa dayalı dönüşüm işlemlerinde en fazla ayrık kosinüs dönüşümü kullanılmaktadır. Blok temelli yöntemlerin en önemli avantajlarından biri düşük bellek ihtiyacı olmasıdır. Dezavantajı ise blok kenarlarının sahip olduğu bozunumlardır. Görüntüye dayalı dönüşümlerin başında ise ayrık dalgacık dönüşümü gelmektedir. Bu yöntem bloğa dayalı yöntemlerin sebep olduğu bozunumu ortadan kaldırmaktadır fakat bellek ihtiyacı problemi ortaya çıkmaktadır. Dönüşüm temelli yöntemlerden en bilinen iki yöntem aşağıda anlatılmaktadır.



### **2.3.1.1. JPEG**

JPEG sürekli tonlu görüntülerin sıkıştırılması için geliştirilen bir yöntemdir. JPEG görüntü sıkıştırma aşamasında dönüşüm nicemleme ve kayıpsız kodlama gibi işlem basamaklarını bir arada kullanır.

JPEG dosyasının oluşturulması birçok aşamadan oluşmaktadır [10]. İlk aşama olarak görüntü bloklara ayrılmaktadır. Görüntü bloklara ayrıldıktan sonra her bir blok ayrı ayrı ayrık kosinüs dönüşümüne sokulur. Dönüşümden geçirilen bloklar nicemleme işlemine sokulur. Nicemleme işlemi kayıplı bir işlem olduğu için JPEG kayıplı bir yöntemdir. JPEG yönteminde ayrık kosinüs dönüşümünün kullanılma sebebi enerji odaklama işlemini iyi yapması ve uygulamanın kolay olmasıdır. Nicemleme işleminden sonra veri zikzak taranarak entropi kodlamaya gönderilir.

### **2.3.1.2. JPEG2000**

JPEG2000 çıkış amacı JPEG yönteminin kısıtlamalarını gidermektir. JPEG2000 sıkıştırma yönteminin temelinde dalgacık dönüşümünü vardır. JPEG sıkıştırma yönteminde yüksek sıkıştırma oranlarına çıkıldığında blok kenar geçişleri belirginleşmektedir. JPEG2000'de blok yapısı olmadığı için yüksek sıkıştırma oranlarında bu kısıt ortadan kaldırılmaktadır. JPEG2000'nin barındırdığı birçok önemli özellik vardır. JPEG2000 sayesinde istenilen bölmeler diğer bölmelere göre daha iyi sıkıştırılabilir. JPEG2000 yöntemi JPEG yönteminin sahip olmadığı kayıpsız sıkıştırma özelliğine de sahiptir [15].

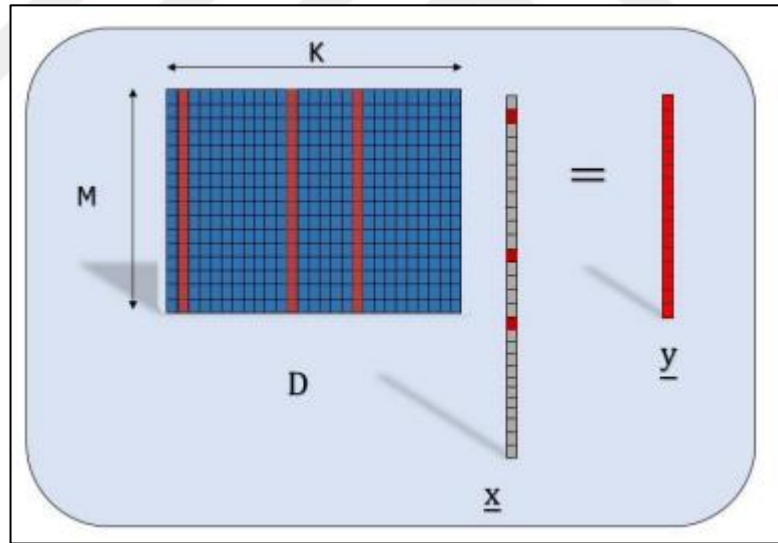
### **2.3.2. Sözlük öğrenme temelli sıkıştırma**

Görüntüleri anlamlı bir şekilde ifade edebilmek için atomlardan oluşan yapıya sözlük denmektedir. Şekil 2.5'te sözlük öğrenme temelli yapının genel sistemi gösterilmektedir. Bu yapıda D ile gösterilen kısım sözlüğü ifade etmektedir. Sözlük üzerinde işaretlenmiş her bir kırmızı sütun sözlüğe ait atomu ifade etmektedir. Sözlük atomlarının belirli bir kısmının kullanılmasıyla veriyi belli bir ölçüde ifade etmektedir [31]. Sözlük öğrenmesi temelli yöntemlerde, bir sözlük ve sözlüğe bağlı katsayı matrisi vardır. Sözlükler çevrim içi öğrenmeye sahip olan ve olmayan olmak üzere ikiye ayrılmaktadır. Çevrim içi öğrenmede sözlük atomları arası değişim belirli

eşiğin altına düşene kadar devam etmektedir [12]. Katsayı matrisindeki elemanlar, çoğunluğu sıfırlardan oluşan bir yapıya sahiptir. Katsayı matrisindeki elemanların sıfırlardan oluşan yapıda olması, katsayı matrisinin seyrek yapıda olmasını sağlamaktadır [29]. Sözlük öğrenmesinde kullanılan yöntemlerde amaç katsayı matrisi ile sözlük çarpımı sonucu oluşan çıkış görüntüsü ile giriş görüntüsü arasındaki karesel hatanın düşük olmasıdır [30]. Bir diğer hedefte seyrekliği en üst seviyede tutabilmektir. Sözlük öğrenmedeki optimizasyon işlemleri iki farklı parametreye göre yapılmaktadır [33].

- Sözlüğün sabit tutulup katsayı matrisinin güncellenmesi
- Katsayılar sabit tutularak sözlüğün güncellenmesi

Pratikte katsayı vektörlerinin bulunması için temel takip algoritması (BP) [20], dik eşleşme takip algoritması (OMP) [18], eşzamanlı dikgen eşleşen algoritma (SOMP) [5] gibi yöntemler kullanılmaktadır.



Şekil 2.5. Sözlük öğrenmesi temelli sıkıştırma yapısı [16]

Sözlüğün güncellenmesi için ise  $K$  adet – Tekil Değer Ayrıştırma (K-Singular Value Decomposition - K-SVD) [17], Optimal Yönler Metodu (“Method of Optimal Directions – MOD”) [18] gibi algoritmalar geliştirilerek sözlük güncellenmektedir. Ayrıca çevrim içi öğrenmeye sahip olmayan sözlük öğrenme metotları da mevcuttur [5]’de anlatılan sözlük öğrenme metodunda verinin genelinden alınan örnekler üzerine PCA dönüşümü uygulanarak sözlük oluşmaktadır. Bu yöntemin avantajı hızlı

olmasıdır çünkü diğer yöntemler her bir güncellemede zaman kaybetmektedirler. Yapılan çalışmalarda PCA dönüşümü spektral ilintiyi en iyi kaldıran ve sıkıştırma performansı en yüksek yöntem olarak öne çıkmaktadır [5,6]. Hem hızlı hem de spektral ilintiyi iyi kaldırdığı için önerilen yöntemde PCA temelli sözlük öğrenme kullanılmıştır.

### 2.3.3. Bölütleme temelli sıkıştırma

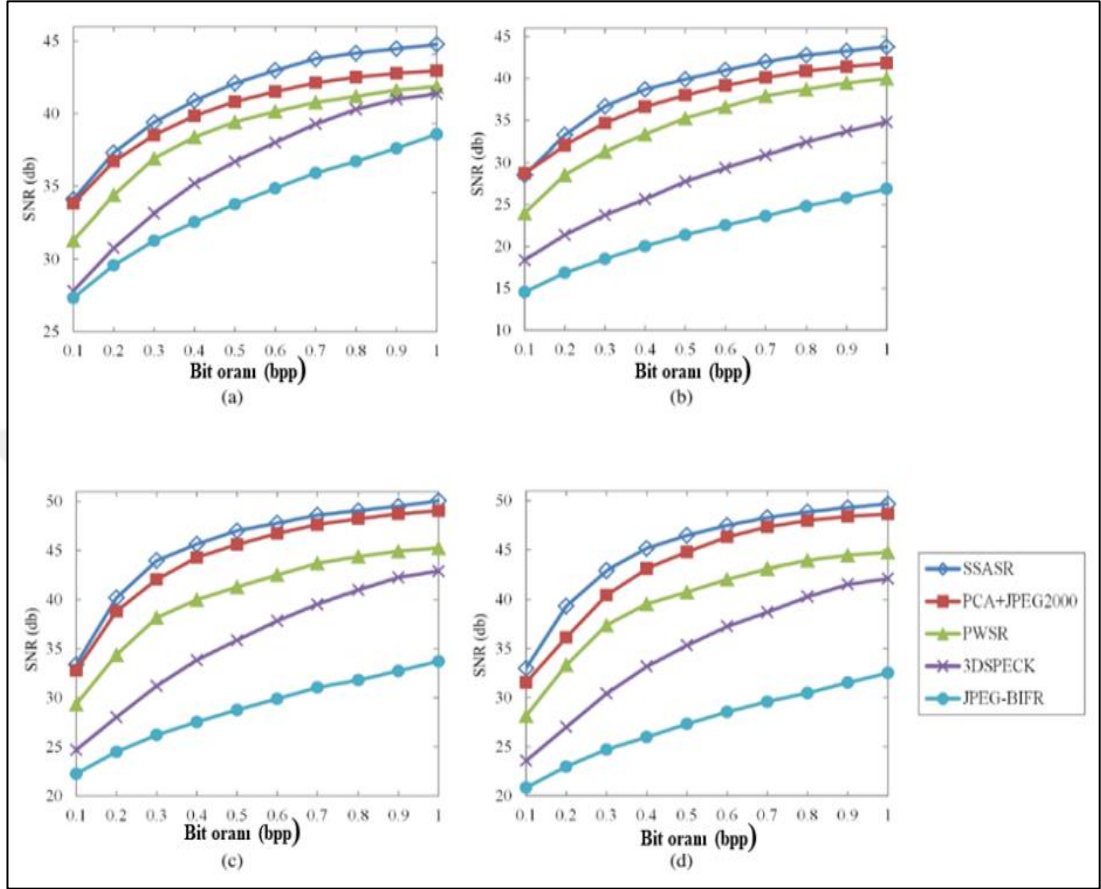
Hiperspektral görüntülerde sıkıştırma işlemleri için bölütleme temelli yöntemler tercih edilmektedir. Bölütleme temelli yöntemlerin tercih edilme sebebi, benzer yapıları bir araya getirerek sıkıştırma performansını artırmaktır. Literatürde hiperspektral görüntüleri bölütlemek için pek çok yöntem önerilmektedir. Bölütleme temelli sıkıştırmanın, en önemli yöntemlerinden biri de süperpiksel tabanlı olan yöntemlerdir.



Şekil 2.6. SLIC süperpiksel bölütleme yöntemi [20]

Şekil 2.6’da SLIC yöntemiyle süperpikselleştirilmiş görüntü görülmektedir. Literatürdeki güncel çalışmalar incelendiğinde süperpiksel temelli kayıplı hiperspektral görüntü sıkıştırmanın pek çok avantaj sağladığı dile getirilmiş ve

sıkıştırma performanslarının da dikkat çekici olduğu görülmüştür [5,6]. [5]'deki yaklaşımın sağladığı avantajlar aşağıdaki şekilde verilmiştir:



Şekil 2.7. SSASR yönteminin sıkıştırma başarımı (a) Indian Pines.(b) WashingtonDC. (c) Moffett. (d) Jasper Ridge için verilmiştir [5]

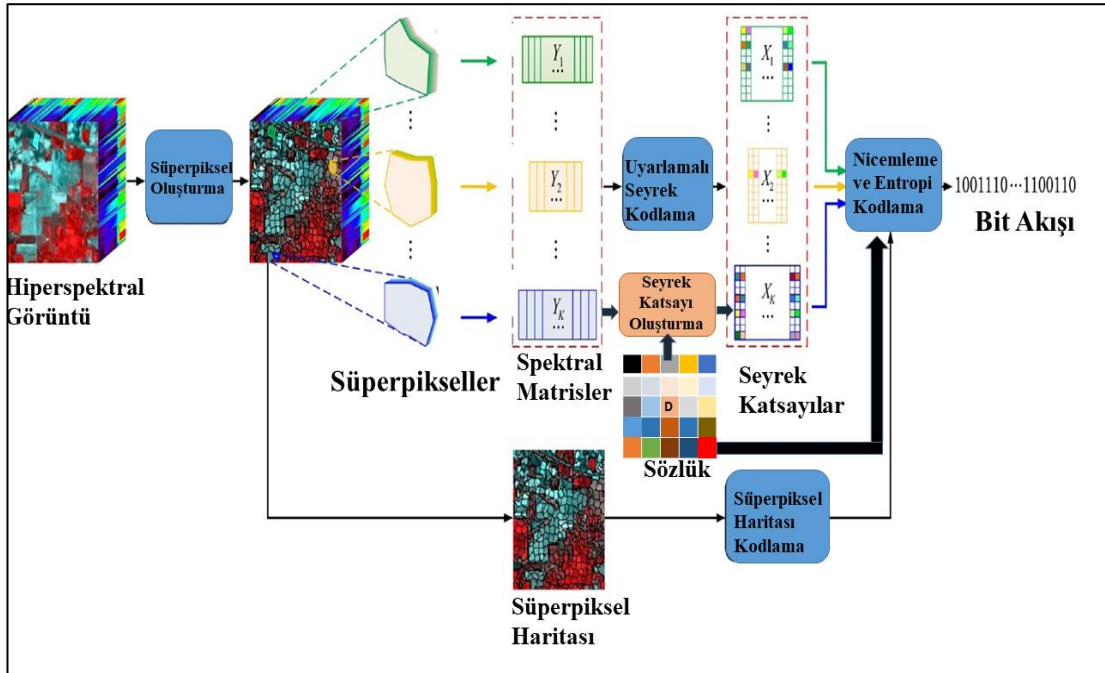
- Süperpikseller, birbirine uzamsal olarak yakın pikseller içerisinde benzer renk ve doku özelliklerine sahip bölgelerdir. Hiperpspektral görüntü için herhangi bir süperpiksel içerisinde bulunan pikseller benzer spektral imzalara sahip olduğu için birbiriyle ilintilidir. Bu özellik kullanılarak sıkıştırma performansı artırılmaktadır.
- Bazı uygulamalarda hedef bölgelerin daha yüksek kalitede saklanması ve arkaplan bölgesinin ise düşük kalitede sıkıştırılması istenebilir. Süperpiksel temelli sıkıştırma yöntemleri ile daha önceden tanımlanan ilgi bölgelerinin (Region of Interest-ROI) farklı sıkıştırma oranlarıyla sıkıştırılabilmesi mümkün olmaktadır. Bu da yönteme bir üstünlük sağlamaktadır.

- Süperpiksel sayısı, görüntü içerisindeki toplam piksel sayısından oldukça düşüktür. Bu yüzden, süperpiksel tabanlı işlemler piksel tabanlı işlemlerden daha verimli olmaktadır.
- Süperpiksel kullanan hiperspektral sıkıştırma yöntemi, literatürdeki diğer yöntemlere göre daha yüksek performanslı bir sıkıştırma sağlayabilmektedir. Şekil 2.7’de paylaşılan karşılaştırmada [5]’deki yöntemin PCA+JP2K, 3D-SPECK ve PWSR yöntemlerinden daha yüksek SNR değerlerine ulaştığı görülmektedir.

Diğer yöntemlerden biri ise genel piksel (General Pixel) tabanlı olan yöntemdir [6]. Bu yöntemde ise veri anlamlı gruplara ayrılırken K-Ortalamalar yöntemini kullanarak gruplama işlemini yapmaktadır. Diğer kümeleme tabanlı yöntemlerden biri ise bulanık kümeleme tabanlı olan yöntemlerdir. Bulanık kümeleme tabanlı olan yöntemlerde hiperspektral görüntülerin bölütlenmesinde kullanılmaktadır. Uydu ile alınan görüntülerin uzamsal çözünürlükleri düşük olduğu için alınan görüntülerde her bir piksele düşen alan büyük olmaktadır. Bu sebeple pikselde saklanan bilgi, farklı cisimleri içinde barındıra bileceği için katışimli bilgidir. Bu piksele etiket atarken bulanık temelli yöntemlerin kullanılması verimli olmaktadır. Bulanık kümeleme algoritmalarına bulanık c-ortalamalar algoritması (BCO) ve bulanık Gustafson-Kessel algoritması (GKK) örnek olarak verilebilir.

### 3. HİPERSPEKTRAL GÖRÜNTÜLERİN SÜPERPİKSEL VE SÖZLÜK ÖĞRENMESİ TEMELLİ SIKIŞTIRILMASI (SSASR)

Süperpikseller benzer spektral özelliğe sahip komşu spektral imzaları bir grup içinde toplamaktadır. Bu sayede görüntü homojen yapılara bölünmektedir. Homojen yapılara bölünen görüntü seyrek gösterim için elverişli hale gelmektedir. Homojen yapıların seyrek gösterimle daha iyi ifade edilme sebebi, şöyle açıklanmaktadır. Seyrek gösterim temelli olan yöntemler, tek bir sözlükle bir grup pikseli ifade etmektedir. Eğer grup içinde bulunan piksellerin birbirleriyle ilintisi düşükse tek bir seyrek katsayı vektörüyle birbirinden uzak olan bu yapılar ifade edilememektedir. Bu sebeple ifade edilecek grup elemanlarının ilintisi yüksek olmalıdır. Süperpikseller elemanları gruplarken hem uzamsal hem de spektral ilintiyi dikkate aldığı için sıkıştırma işleminde seyrek katsayılarla gösterimi güçlü olmaktadır. Bölüm 2’de süperpiksel temelli olan SSASR [5] yönteminin diğer yöntemler karşısında başarımı gösterilmektedir.



Süperpiksel temelli sözlük öğrenmesi ile sıkıştırma işlemi hiperspektral görüntüler üzerinde gerçekleştirilirken SSASR yönteminde Şekil 3.1'deki aşamalar takip edilmiştir ve aşağıda bu aşamalar detaylı olarak anlatılmıştır.

- Görüntü ilk aşamada  $K$  adet süperpiksele ayrılmaktadır. Buradaki  $K$  değeri her bir görüntü için, görüntünün içerdiği piksel sayısına da bağlı olarak farklı olabilmektedir.  $K$  değerinin farklı değerlerde seçilmesinin avantaj ve dezavantajları deneysel çalışmalar kısmında tartışılmıştır
- Elde edilen  $K$  adet süperpiksel vektörleştirme işlemiyle matris haline çevrilmektedir. Bu işlem sonucunda  $B \times n_k$  boyutlarında matrisler elde edilmektedir. Buradaki  $B$  spektral bant sayısını,  $n_k$  ise  $k$ . süperpikselde bulunaneleman sayısını ifade etmektedir. Vektörleştirme işleminde süperpiksel içerisindeki ilk elemandan başlayarak yukarıdan aşağıya doğru bir tarama yolu izlenir. Bu tarama ile elde edilen spectral imzalar yan yana eklenerek her bir süperpiksel için bir matris oluşturulur ve vektörleştirme işlemi tamamlanır.
- Elde edilen spektral matrislerden seçilen belli sayıda örnek  $K$ -ortalama kümeleme algoritması ile kümelenebilir. Kümelenen verilere ayrı ayrı temel bileşenler analizi (Principal Components Analysis-PCA) uygulanıp sözlük için kullanılacak katsayılar bulunmaktadır. Her bir kümeden bulunan en değerli katsayılardan belli sayıda alınıp sözlüğün atomları oluşturulmaktadır.
- Elde edilen spektral matrisler ve sözlük, eş zamanlı dikgen eşleşen arama (Simultaneous Orthogonal Matching Pursuits-SOMP) algoritmasına giriş olarak uygulanıp, her bir süperpiksel için seyrek katsayılar hesaplanmaktadır.
- Elde edilen seyrek katsayılar, sözlük ve süperpiksel haritası entropi kodlama işlemi ile kodlanarak sıkıştırma işlemi gerçekleştirilmektedir.
- Süperpiksel haritası ayrıyeten kodlanmıştır.

### 3.1. Süperpiksel Bölütleme

Süperpiksellerin bulunmasında basit doğrusal yinelemeli kümeleme (Simple Linear Iterative Clustering-SLIC) tercih edilmiştir. Bu yöntemin tercih edilme sebebi, mevcut yöntemlerden daha hızlı ve daha az bellek ihtiyacı duymasıdır. Ayrıca bölütleme algoritmalarının performansını artıran bir yapıya da sahiptir [20].

Bu yöntem iki önemli özelliğe sahiptir:

- Optimizasyondaki mesafe hesaplamasını, arama alanını süper piksel boyutu ile orantılı bir bölgeye sınırlandırarak, önemli ölçüde azaltır.
- Ağırlıklı mesafe ölçüsü olarak, renk benzerliğini ve uzamsal yakınlığı birleştirerek, aynı anda süperpiksellerin boyutu ve bütünlüğü (compactness) üzerinde kontrol sağlar.

Süperpikseller görüntüyü farklı büyüklükte homojen alanlara böler ve bu alanlar birbiriyle örtüşmemektedir. Süperpiksellerin örtüşmemesi Denklem 3.1’de aşağıdaki şekilde tanımlanmaktadır.

$$\forall i, j \in \left\{ \begin{array}{l} Y_i \cap Y_j = \emptyset \\ Y_i \cup Y_j = Y \end{array} \right. \quad (3.1)$$

Bu eşitlikte  $Y_k = \{y_{k,1}y_{k,2} \dots y_{k,n_k}\}$   $\left( \sum_{k=1}^K n_k = mn \right)$ ,  $Y$  imgesindeki  $n_k$  elemanlı  $k$ . süperpikseli göstermektedir. Buradaki  $K$ , görüntüdeki toplam süperpiksel sayısını ifade etmektedir. Görüntüyü süperpiksellere ayırırken HSI pikselleri arasındaki ilişki, spektral imzalar arasındaki ilişki olarak değerlendirilmektedir. Uzaklık ifadesi Denklem 3.2’deki gibi bulunmaktadır.

$$d(y_a, y_b) = \left\| \frac{y_a}{\|y_a\|_2} - \frac{y_b}{\|y_b\|_2} \right\|_2 \quad (3.2)$$

Burada  $y_a$  ve  $y_b$  herhangi bir spektral imzayı ifade etmektedir  $d(y_a, y_b)$ ,  $y_a$  ve  $y_b$  arasındaki mesafeyi ifade etmektedir. Spektral imzalar arasındaki uzaklığa normalizasyon işlemi uygulanarak süperpiksel oluşturmadaki etkisi belli aralığa çekilmektedir. Süperpiksellerin benzerliğini hesaplariken ayrıca seçilen pikselin küme merkezlerine olan Öklid uzaklığına bakılmaktadır. Son olarak genel uzaklık ifadesi, yukarıda tanımlanan iki uzaklık ifadesinin belirli oranlarda birleştirilmesi ile elde



edilmektedir. Süperpikseller oluşturulurken takip edilen işlem basamakları Tablo 3.1’de verilmektedir.

Tablo 3.1. Süperpiksel oluşturma

---

Giriş Verileri: 1) Hiperspektral görüntü ( $M \times N \times B$  boyutunda), 2) Süperpiksel sayısı  $K$ , 3) İterasyon sayısı  $I$ ;

Çıkış Verileri: Süperpiksel etiket haritası  $L$  ( $M \times N$  boyutunda);

---

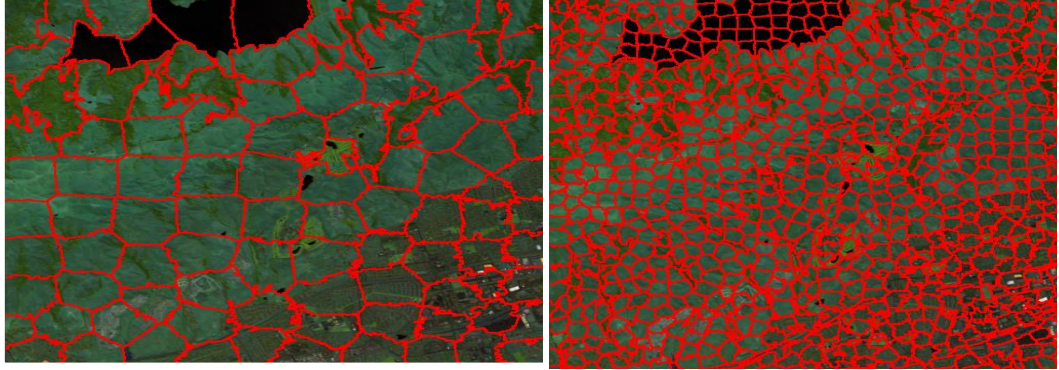
- 1: İterasyon sayısı  $q = 0$  ve süperpiksel etiket haritası  $L = 0$ ;
- 2:  $K$  adet merkezin başlangıç değerini eşit aralıklı seçip ata;
- 3: Pencere genişliğini hesapla  $2\sqrt{\frac{MN}{K}} \times 2\sqrt{\frac{MN}{K}}$
- 4: Döngü 1 İterasyon sayısı aşılmayana kadar devam et ( $q < I$ ):
- 5: Döngü 2 Her bir  $y$  piksel vektörü için şu işlemleri yap:
- 6: Her bir pencere içindeki pikselin pencere içindeki merkezlere olan uzaklığını hesapla;
- 7: Minimum uzaklığı veren merkez etiketini  $L(y)$ 'e ata;
- 8: Döngü 2 sonu
- 9:  $K$  adet küme merkezlerindeki vektörleri güncelle;
- 10: İterasyon sayısını bir artır  $q = q + 1$
- 11: Döngü 1 sonu

---

Giriş bilgisi olarak ( $M \times N \times B$ ) boyutlarında hiperspektral görüntü,  $K$  adet süperpiksel sayısı ve maksimum iterasyon sayısı verilir. Görüntünün yatay ve dikey büyüklüğü boyutlarında, tamamı sıfırlardan oluşan süperpiksel etiket haritası oluşturulur. Başlangıç iterasyon sayısı 0 yapılır.

Eşit aralıklarla  $K$  adet küme merkezi seçilir. Seçilen küme merkezi sayısına bağlı olarak arama yapılacak pencere boyutu hesaplanır. Her bir pikselin, seçilen pencere boyutunun dört katı uzaklık içindeki küme merkezlerine olan uzaklığı hesaplanarak, hangi küme merkezine daha yakın ise o küme merkezine ataması yapılır. Süperpiksel haritası güncellenir ve bu işlem iterasyon sayısı kadar tekrar edilir. Şekil 3.2’de, Moffett görüntüsünde 100 ve 1000 adet süperpiksel sayısı için SLIC ile elde edilen süperpiksel haritasıları görülmektedir. Süperpiksel haritalarına bakıldığı zaman 1000

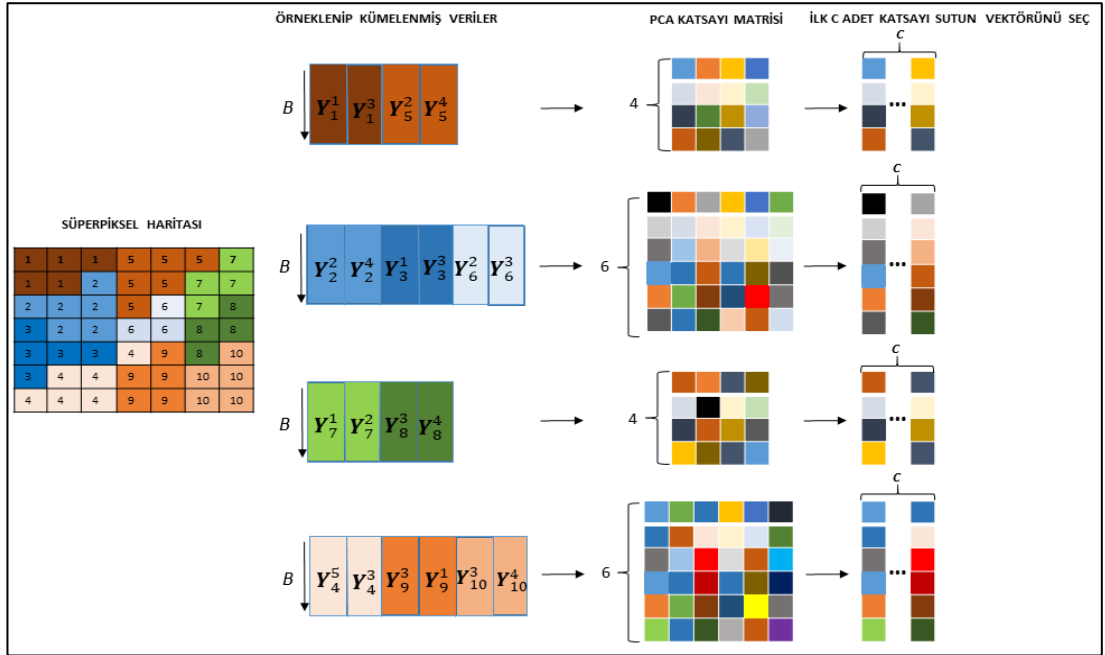
adet süperpiksele sahip olan harita bölge sınırlarına daha kompakt bir şekilde yerleşmektedir.



Şekil 3.2. Moffett görüntüsünde 100 ve 1000 adet süperpiksel için süperpiksel haritası

### 3.2. Sözlük Öğrenmesi

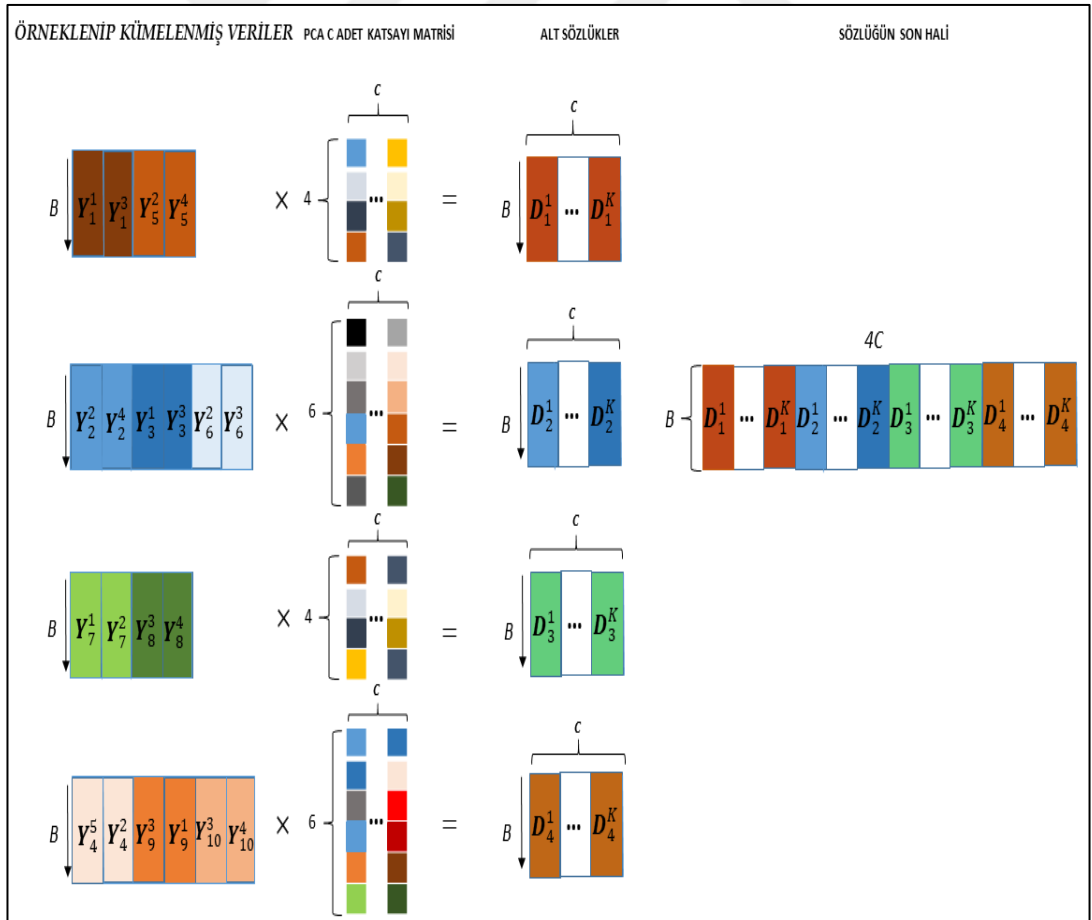
Sözlük öğrenmesi işleminde K-ortalama kümeleme temelli bir işlem ve PCA kullanılmaktadır [21]. İlk olarak K adet süperpikselin her birinden T adet piksel seçilir. Her bir süperpikselden seçilen örnekler uç uca eklenerek ayrı ayrı sütun vektörü haline getirilir.



Şekil 3.3. Sözlük oluşturmada süperpiksellerin gruplanması ve PCA katsayılarının bulunması

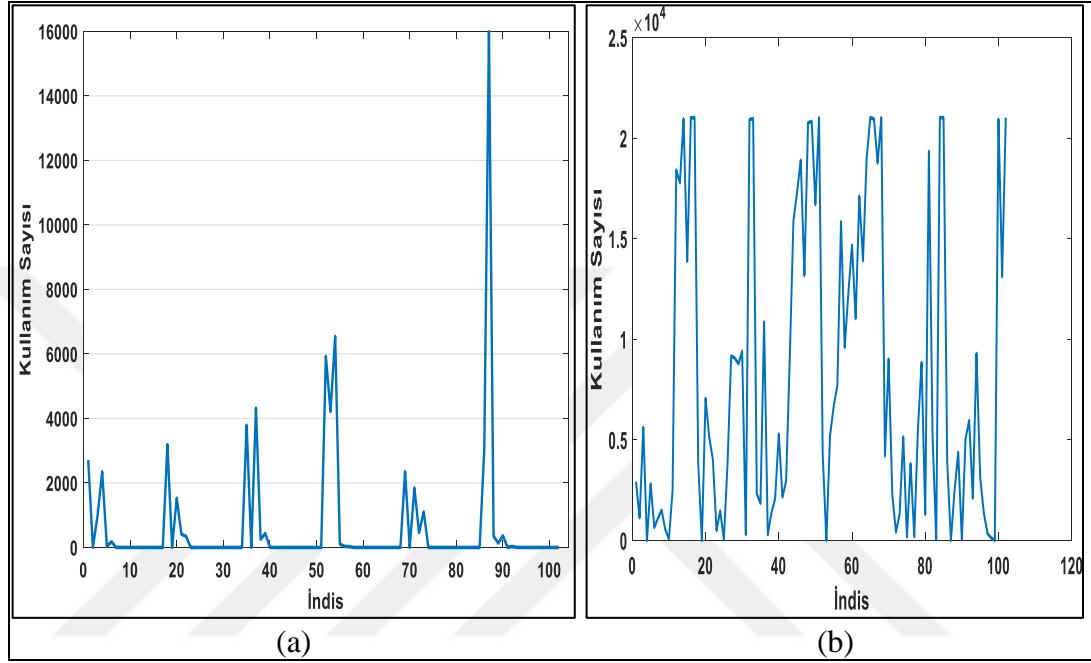
Sütun vektörleri birleştirilerek  $S \times K$  boyutlarında örnek seçim matrisi oluşturulur.  $S = B \times T$  boyutunda olmaktadır. Burada  $B$  bant sayısı,  $T$  seçilen örnek sayısıdır. Oluşturulan örnek seçim matrisine  $K$ -ortalama kümeleme algoritması uygulanarak veriler gruplandırılmaktadır. Gruplanan verilere PCA uygulanarak her bir grup için hesaplanan özvektörlerden en değerli  $C$  adet öz vektör seçilir. Bu seçim işlemi Şekil 3.3'te gösterilmiştir [32].

Gruplanan veriler, seçilen bu özvektörler ile çarpılır ve sözlük için atomlar öz uzayda oluşturulur. Şekil 3.4'te alt sözlüklerin oluşturulması ve birleştirilmesi gösterilmiştir. Daha sonra sözlük atomları birim vektör haline dönüştürülerek normalize edilir. Oluşturulan sözlük seyrek katsayı hesabında kullanılır. Seyrek katsayı hesabından sonra sözlükte güncellemeye gidilir. Sözlük atomlarından seyrek katsayı hesabında belli sayıdan az kullanılanlar atılır ve gerçek seyrek katsayı hesabı yeni elde edilen bu sözlük üzerinden yürütülür.



Şekil 3.4. Sözlük oluşturmada alt sözlüklerin bulunması ve sözlüğün son hali

Düşük SNR oranlarında sözlüğün atomları az sayıda kullanıldığı için, veri sıkıştırma büyük avantaj sağlamaktadır. Şekil 3.5(a)'da 0,1 bpp için sözlük atomlarının kullanım sayıları görülmektedir. Yukarıda belirtildiği gibi 0,1 bpp için grafikte çok az sayıda sözlük atomu kullanıldığı görülmektedir. Şekil 3.5(b)'de ise 1 bpp için çok sayıda sözlük atomu kullanıldığı görülmektedir.



Şekil 3.5. (a) Sözlük atomlarının 0,1 bit için kullanım durumu (b) Sözlük atomlarının 1 bit için kullanım durumu

### 3.3. Seyrek Katsayıların Belirlenmesi

Seyrek katsayı tanımlama işleminde SOMP algoritması kullanılmıştır. SOMP algoritmasının kullanılmasının sebebi [21] ve [22]'deki çalışmalarda detaylı olarak açıklanmıştır. Bu çalışmalarda, genellikle komşu olan ve benzer spektral imzaları paylaşan HSI piksellerinin eşzamanlı temsilinin seyrek tanımlama performansını artırdığına değinilmiştir.

Süperpikseller ilintili pikselleri gruplandığı için, seyrek katsayı tanımlama modelinde eşzamanlı olarak tanımlanabilmektedir.  $Y_k$ ,  $k$ . süperpikselin spektral imza matrisini gösteriyorsa  $k$ . süperpikselin ayık tanımlanması Denklem 3.3'de ifade edilmektedir.

$$\min \|X_k\|_{\text{row},0} \quad \text{s.t. } Y_k = DX_k + N \quad (3.3)$$

Denklem 3.4'te  $X_k$  ayrık katsayı matrisini göstermektedir,  $\min \|X_k\|_{\text{row},0}$  ise  $X_k$ 'daki sıfır olmayan satır sayısını,  $D$  sözlüğü,  $N$  ise gürültüyü ifade etmektedir. Sözlüğümüz  $D$  olmak üzere seyrek katsayılar aşağıdaki uyarlama problemine göre bulunmaktadır.

$$X_k = \operatorname{argmin} \|X_k\|_{\text{row},0} \quad \text{s.t. } \|Y_k - DX_k\| \leq \delta_k \quad (3.4)$$

Denklem 3.4'teki  $\delta_k$ ,  $k$ 'nıncı süperpiksel için hedef hatayı ifade etmektedir. Her bir süperpiksel için farklı hedef hataları tanımlayarak ilgili süperpikselin farklı kalite değerlerinde sıkıştırılması mümkün olmaktadır. Fakat bu çalışmada hedef veri bozulması için oran-bozunum değerinin en uygun seviyeye çekilmesi hedeflenmektedir. Bit oranı, doğrudan sıfırdan farklı olan seyrek katsayı miktarı ile pozitif ilişkilidir. Sıfır olmayan katsayı miktarı arttıkça bit oranı artmakta ve görüntüyü geri elde etme iyileşmekte; sıfır olmayan katsayı miktarı azaldıkça bit oranı azalmakta ve buna bağlı olarak SNR düşmekte, görüntüyü geri elde etme kötüleşmektedir.

$$\text{Rate } \alpha \sum n_k \|X_k\|_{\text{row},0} \quad (3.5)$$

$n_k$   $k$ 'nıncı süperpikseldeki piksel sayısını ifade ettiği taktirde,  $k$ . süperpikseldeki toplam sıfır olmayan eleman sayısı  $\sum_k n_k \|X_k\|_{\text{row},0}$  şeklinde ifade edilebilmektedir. Bit oranını minimize etmek istendiğinde,  $\sum_k n_k \|X_k\|_{\text{row},0}$  işleminin sonucunun minimize edilmesi gerekmektedir.  $s_k$ ,  $k$ . süperpikseldeki toplam sıfır olmayan eleman sayısıdır ve  $s_k = \sum_k n_k \|X_k\|_{\text{row},0}$  ile gösterilir.  $\delta_k(s_k)$   $s_k$ 'nın bir fonksiyonu olarak Denklem 3.5'deki optimizasyona göre hedef hatayı temsil etmektedir. Verilen genel hedef hatayı en küçükleme problemi aşağıdaki gibi tanımlanmaktadır.

$$\min \delta_k \sum_{k=1}^K s_k, \quad \left| \sum_{k=1}^K \delta_k (s_k - \delta_t) \right| < \check{S} \quad (3.6)$$

Denklem 3.6'daki  $\check{S}$ , toplam hedef hata ile  $s_k$ 'nın fonksiyonuna bağlı hedef hata arasındaki tolerans hatayı göstermektedir ve çok küçük bir değere sahiptir. Genellikle

farklı süperpikseller farklı seyrek tanımlama performansına sahiptir. Yani, farklı  $k$  değerleri için  $\delta_k(s_k)$ 'nın farklı fonksiyon eğrilerine sahip olmasına karşılık gelmektedir ve bu sebeple en uygun çözüme yaklaşmak için açgözlü algoritma temelli ayrık kodlama yöntemi önerilmiştir.

Çözüm için ilk olarak bütün süperpikseller büyük hedef hatası ile tanımlanır. Sonra

genel hata  $\sum_{k=1}^K \delta_k(s_k)$ , hedef hatanın ( $\delta_k$ ) maksimum iterasyon sayısına ( $I$ )

bölünmesiyle elde edilen hata azaltma adım genişliği doğrultusunda her bir adımda azaltılır. Her bir çevrimde Denklem 3.7'de verildiği gibi, hangi süperpiksel hedef hatasındaki değişime bağlı olarak daha az katsayı arttırıyorsa, o süperpiksel Denklem 3.8'de verilen eşitlik doğrultusunda seçilir. Bu seçim standart açgözlü algoritma ile yapılır [23].

$$\frac{\Delta s_k^i}{\Delta \delta_k^i} = \frac{s_k^{i-1} - s_k^i}{\delta_k^{i-1} - \delta_k^i} \quad (3.7)$$

$$\hat{k}^i = \operatorname{argmin}_k \frac{\Delta s_k^i}{\Delta \delta_k^i} \quad (k=1,2,3,\dots,K) \quad (3.8)$$

Seçilen süperpiksel daha küçük hedef hatayla tanımlanır ve bu sayede toplam hata azaltılmış olur. Son olarak,  $\left| \sum_{k=1}^K \delta_k(s_k - \delta_t) \right| < \check{S}$  durumu sağlanınca iterasyon durdurulur ve seyrek tanımlama işlemi tamamlanmış olur. Yukarıda detayları verilen adaptif seyrek kodlamanın işleyişi Tablo 3.2'de verilmektedir.

Tablo 3.2'de verilen denklemlere baktığımızda seyrek kodlamanın işlem adımlarını sırasıyla gösterilmektedir. Genel hata hesaplama işleminin bulunması. Hedef hata değerine göre başlangıçta hesaplanan seyrek katsayı değerlerinin hesaplanma işlemi. Hesaplanan seyrek katsayıların değerlerinin her bir döngü iterasyonu sonunda sahip olduğu yeni güncel değerlerin hesaplanma aşamaları adım adım gösterilmektedir.

Tablo 3.2. Adaptif seyrek kodlama

---

Giriş Verileri: 1) Süperpikseller  $\{Y_k\}$   $k = 1,2,3,\dots,K$  2). Eğitimli sözlük  $D$  ;  
3) Genel hedef hata  $\delta_t$  ; Çıkış Verileri: Seyrek katsayılar  $\{X_k\}$   $k = 1,2,3,\dots,K$

---

1:Adım genişliğini hesapla  $\delta_t/I$ ;

2:her bir süperpikselin başlangıç hedef hatasını ve ikinci hedef hatasını ata  
 $\delta_k^0 = 2\delta_t n_k / MN$ ,  $\delta_k^1 = \delta_0 - \Delta\delta$  ifadeye göre seyrek katsayıları hesapla  $X_k^0$ ,  $X_k^1$  ;

3:Başlangıçta sıfır olmayan katsayıları adım ikideki birinci ve ikinci hedef hata için hesapla  $s_k^0 = n_k \|X_k^0\|_{row,0}$ ,  $s_k^1 = n_k \|X_k^1\|_{row,0}$

4: Denklem 3.7’de verilen denkleme göre  $\frac{\Delta s_k^1}{\Delta \delta_k^1}$  oranını hesapla.

5: Seyrek katsayıları ilgili matrise ata ve  $i$ ’ yi bir artır  $X_k = X_k^0$  ve  $i = i + 1$ ;

6: Döngü 1  $\left| \sum_{k=1}^K \delta_k (s_k - \delta_t) \right| < \check{S}$  eşitlik sağlanana kadar devam et

7:  $\hat{k}^i$  değişkenini Denklem 3.8’de verilene göre bul

8: Çöz  $X_{\hat{k}^i}^i = \operatorname{argmin} \|X_{\hat{k}^i}^i\|_{row,0}$  s.t.  $\|Y_{\hat{k}^i} - DX_{\hat{k}^i}^i\|_F^2 < \delta_{\hat{k}^i}^i$

9: Yenile  $\delta_{\hat{k}^i}^1 = \delta_{\hat{k}^i}^i - \Delta\delta$   $s_{\hat{k}^i}^i = n_{\hat{k}^i} \|X_{\hat{k}^i}^i\|_{row,0}$

10: Çöz  $X_{\hat{k}^i}^{i+1} = \operatorname{argmin} \|X_{\hat{k}^i}^i\|_{row,0}$  s.t.  $\|X_{\hat{k}^i}^i - DX_{\hat{k}^i}^{i+1}\|_F^2 < \delta_{\hat{k}^i}^{i+1}$

11: Yenile  $s_{\hat{k}^i}^{i+1} = n_{\hat{k}^i} \|X_{\hat{k}^i}^{i+1}\|_{row,0}$

12: Denklem 3.7’de verilen eşitliğe göre  $\frac{\Delta s_{\hat{k}^i}^{i+1}}{\Delta \delta_{\hat{k}^i}^{i+1}}$  oranını hesapla.

13:  $X_{\hat{k}^i} = X_{\hat{k}^i}^i$  ve sonra  $i = i + 1$ ;

14: Döngü 1 sonu

---

### 3.4. Nicemleme, Entropi Kodlama ve Kod Çözme

Bu kısımda, seyrek gösterimi elde edilen verinin sıkıştırılmasında tanımlı işlem aşamaları verilmektedir. İlk aşamada yapılan işlem, sıfır olmayan seyrek katsayı değerleri, sıfır olmayan seyrek katsayıların satır indisleri, süperpiksel haritası ve sözlüğün kodlanmasıdır. Entropi kodlama aşamasında Huffman kodlama kullanılmıştır [24]. Sıfır olmayan seyrek katsayılar ve sözlük değerleri tamsayı olmadığından, Huffman kodlama öncesinde nicemleme işlemi gerçekleştirilmektedir.

Bu aşamada tekdüze nicemleyici kullanılmıştır. Veriyi geri elde etme aşamasında Huffman kod çözücü kullanılmış ve elde edilen seyrek katsayılarla sözlük atomları çarpılıp geri elde etme işlemi tamamlanmıştır. Kod çözme işlem aşamaları Tablo 3.3'te verilmektedir.

Tablo 3.3. Kod çözme

---

Giriş Verileri: 1) Seyrek katsayıların bit akışı, 2) Kodlanmış süperpiksel haritasının bit akışı, 3) Sözlüğün bit akışı;
Çıkış Verileri: Geri elde edilmiş hiperspektral görüntü $\hat{Y}$ ;
1: Huffman kod çözücü tarafından bit akışı çözülür;
2: $\hat{X}$ ve sözlük $D$ ters nicemlenerek seyrek katsayılar elde edilir;
3: Süperpiksel haritasının çözülmesi ve $k$ . süperpiksel için uzamsal koordinatların $(\varphi_k)$ bulunması;
4: Döngü 1 $k=1$ den $K$ süperpiksel sayısına kadar
5: $k$ . süperpiksel için spektral imzaları hesapla $\hat{Y}(\varphi_k) = D \cdot \hat{X}_k$
6: Döngü 1 sonu

---



#### 4. SÜPERPİKSEL VE SEYREK GÖSTERİM TEMELLİ SIKIŞTIRMA İÇİN ÖNERİLEN YÖNTEMLER

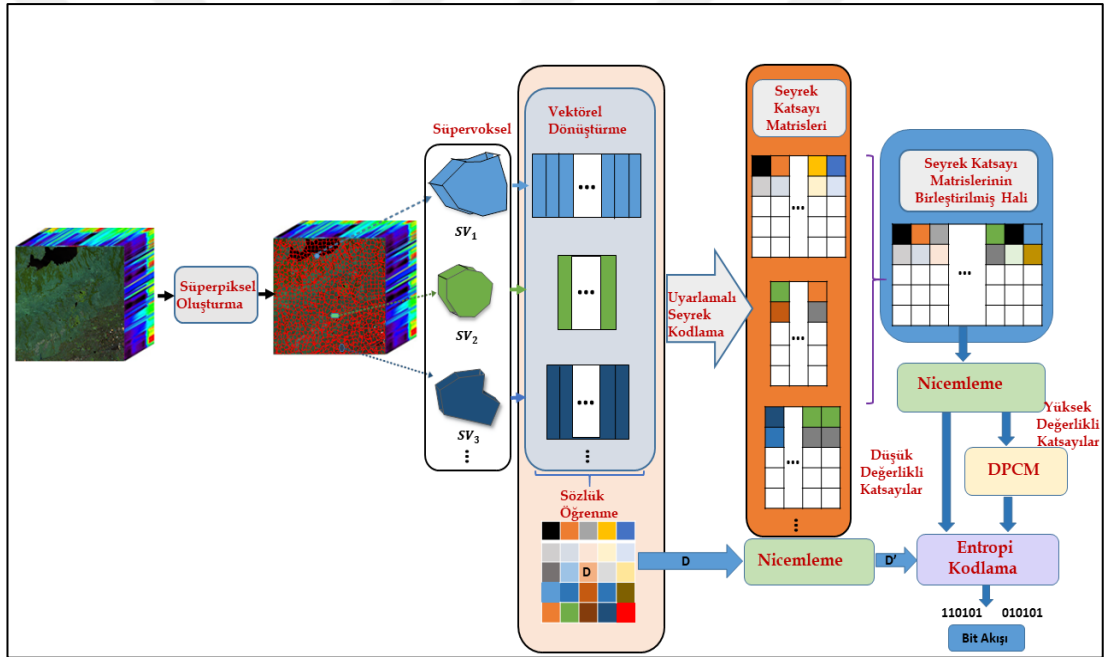
Literatürdeki güncel çalışmalar incelendiğinde süperpiksel temelli kayıplı hiperspektral görüntü sıkıştırmanın pek çok avantaj sağladığı dile getirilmiş ve sıkıştırma performanslarının da dikkat çekici olduğu görülmüştür [5,6]. [5]'deki yaklaşımın sağladığı avantajlar aşağıdaki şekilde verilmiştir:

- Süperpikseller, birbirine uzamsal olarak yakın pikseller içerisinde benzer renk ve doku özelliklerine sahip bölgelerdir. Hiperspektral görüntü için herhangi bir süperpiksel içerisinde bulunan pikseller benzer spektral imzalara sahip olduğu için birbirleriyle ilintilidir. Bu özellik kullanılarak sıkıştırma performansı artırılmaktadır.
- Bazı uygulamalarda hedef bölgelerin daha yüksek kalitede saklanması ve arkaplan bölgesinin ise düşük kalitede sıkıştırılması istenebilir. Süperpiksel temelli sıkıştırma yöntemleri ile daha önceden tanımlanan ilgi bölgelerinin (Region of Interest-ROI) farklı sıkıştırma oranlarıyla sıkıştırılabilmesi mümkün olmaktadır. Bu da yöntemle bir üstünlük sağlamaktadır.
- Süperpiksel sayısı, görüntü içerisindeki toplam piksel sayısından oldukça düşüktür. Bu yüzden, süperpiksel tabanlı işlemler piksel tabanlı işlemlerden daha verimli olmaktadır.
- Süperpiksel kullanan hiperspektral sıkıştırma yöntemi, literatürdeki diğer yöntemlere göre daha yüksek performanslı bir sıkıştırma sağlayabilmektedir.

SSASR yöntemi süperpiksel temelli kayıplı sıkıştırma yöntemi olup, önerilen çalışmalarda bu yöntem referans alınmaktadır. Önerilen yöntemler sayesinde sıkıştırma performansı ve anomali koruma performansı artırılmıştır. SSASR yöntemi süperpiksel haritasını fazladan kodlamaktadır önerilen yöntemler bunu kaldırmıştır. SSASR yöntemi sözlük oluştururken çoklu PCA kullanmaktadır bu sebeple oluşturduğu sözlük veriyi iyi ifade edememektedir. Yeni geliştirilen yöntemler sayesinde çoklu PCA'yın olumsuz ve eksik yönlerini ortadan kaldırmaktadır.

#### 4.1. Modifiye Edilmiş Sözlük ve Seyrek Gösterimle Süperpiksel Temelli Sıkıştırma Yöntemi (MSSASR)

Önerilen yöntem, SSASR temelli olup bu yönteme birçok yenilik katmaktadır. İlk olarak, SSASR yöntemindeki sözlük öğrenmesi adımı güncellenmiştir. Bu sayede, her bir süperpikselin seyrek katsayılarının daha kompakt olmasını sağlamaktadır. İkinci olarak elde edilen seyrek katsayıların diziliminde yeni bir metot oluşturulmuştur. Bu yenilik, süperpiksel haritasının kodlama ihtiyacını ortadan kaldırmaktadır. Son olarak yüksek genlik değerlerine sahip seyrek katsayılara entropi kodlama öncesi DPCM uygulamaktadır. Bu sayede ekstra bit yükünü ortadan kaldırmaktadır. Önerilen yöntemin genel şeması Şekil 4.1'de verilmiştir.



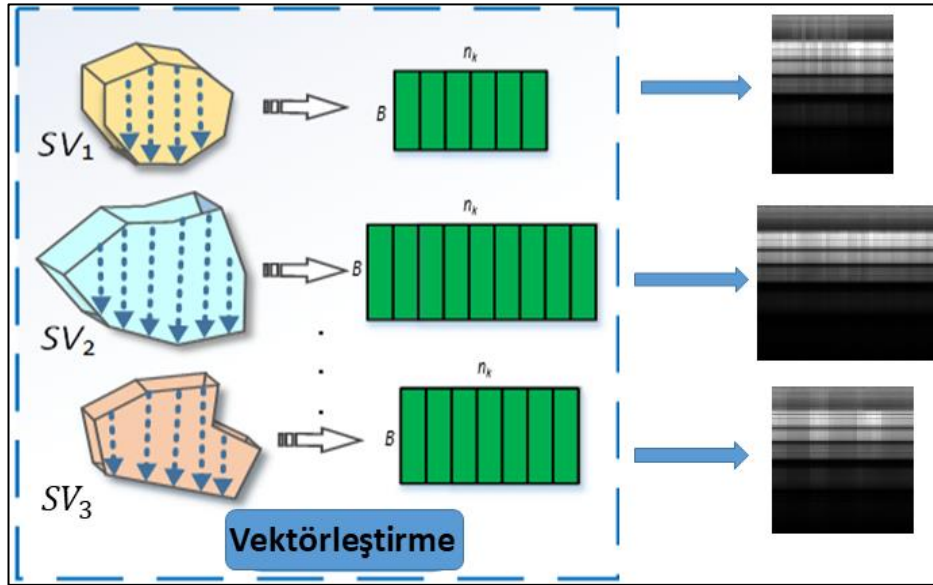
Şekil 4.1. Yöntemin genel işleyişi

Önerilen yöntemin genel akışı şöyle tanımlanmaktadır. İlk olarak görüntüde süperpikseller bulunmaktadır. Her bir süperpiksele vektörleştirme işlemi uygulanmaktadır. İkinci aşamada her bir süperpikselden rastgele seçilen örnek piksellere temel bileşenler analizi (Principal Components Analysis-PCA) uygulanarak sözlük oluşturulmaktadır. Sonraki aşamada her bir süperpiksele karşılık gelen seyrek katsayılar bulunmaktadır. Son aşamada sözlük ve seyrek katsayı satır bilgileri nicemlenerek, entropi kodlanmaktadır. Ayrıca, seyrek katsayılar

nicemlenerek, yüksek dereceli katsayılar DPCM işlemine tabii tutulup düşük dereceli katsayılar ise doğrudan entropi kodlanmaktadır. Yöntemin işlem basamakları aşağıda detaylı olarak açıklanmaktadır.

#### 4.1.1. Süperpiksel oluşturma

Süperpiksel, birbirine benzeyen komşu piksellerin bir araya gelerek oluşturduğu sınırlı alana sahip anlamlı bölgedir. Bu alan veri büyüklüğüne ve süperpiksel sayısına bağlı olmaktadır. Süperpiksel yöntemi veri sıkıştırma, derinlik kestirimi, görüntü segmentasyonu, iskeletleştirme, vücut modeli kestirimi ve obje lokalizasyonu gibi uygulamalarda kullanılmaktadır. Süperpiksel bölütleme sonucunda, piksel tabanlı işlemler yerini süperpiksellere bırakmaktadır. Bu sayede, süperpiksel başına düşen gelecek adımlardaki işlem yükü azalmaktadır. Önerilen yöntemde, veriyi süperpiksellere ayırmak için basit doğrusal yinelemeli kümeleme (Simple Linear Iterative Clustering- SLIC) algoritması tercih edilmektedir [20]. SLIC algoritması işlem yükü daha az, süperpiksel bölütleri daha kompakt ve hızlı olduğu için tercih edilmektedir. Önerilen yöntem içinde SLIC algoritması tercih edilmiştir. Bölüm 3'te SLIC algoritması anlatıldığı için burada anlatılmamıştır.



Şekil 4.2. Vektörleştirme işlemi

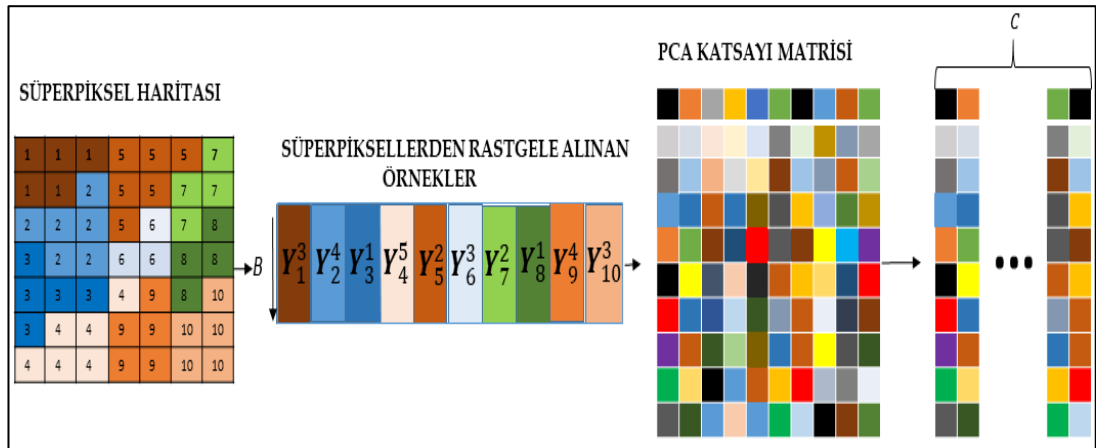
SLIC algoritması yardımıyla elde edilen  $K$  adet süperpiksel Şekil 4.2'de gösterildiği gibi vektörleştirme işlemiyle matris haline dönüştürülmektedir. Bu işlem sonucunda,

$B \times n_k$  boyutlarında matrisler elde edilmektedir. Buradaki  $B$  spektral bant sayısını,  $n_k$  ise  $k$ . süperpikselde bulunan eleman sayısını ifade etmektedir. Vektörleştirme işleminde süperpiksel içerisindeki ilk elemandan başlayarak yukarıdan aşağıya doğru bir tarama yolu izlenir. Bu tarama ile elde edilen spectral imzalar yan yana eklenerek her bir süperpiksel için bir matris oluşturulur ve vektörleştirme işlemi tamamlanır. Şekil 4.2’de bulunan  $SV_1, SV_2, SV_3$  ile gösterilen elemanlar süperpiksellere ifade etmektedirler. Süperpikseller, matrisler halinde işleme girmek için uygun olmadığından, vektörel dönüşüme tabi tutulmaktadır.

#### 4.1.2. Sözlük oluşturma

##### 4.1.2.1. Temel sözlük işlemleri

Sözlük öğrenmesi işleminde PCA yöntemi kullanılmaktadır. İlk olarak,  $K$  adet süperpikselin her birinden  $T$  adet piksel seçilmektedir. Her bir süperpikselden seçilen  $T$  adet örnek Şekil 4.3’te verildiği gibi yan yana dizilerek birleştirilmektedir. Birleştirme işlemi sonrası oluşturulan matrise, PCA uygulanarak temel bileşen katsayı matrisi oluşturulmaktadır.

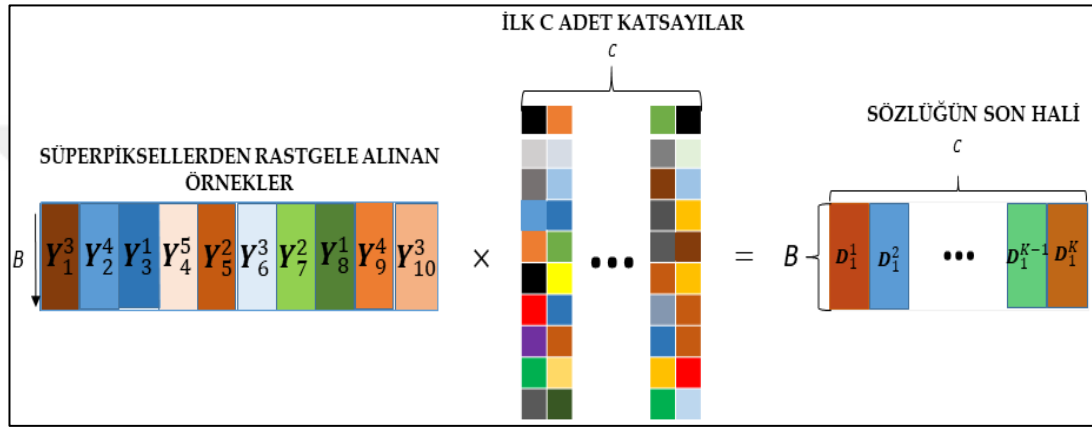


Şekil 4.3. Süperpikselden örnek seçimi ve PCA katsayı matrisinin oluşumu

PCA işlemi sonucu hesaplanan öz vektörlerden en değerli  $C$  adet öz vektör seçilmektedir. Burada  $C$ , sözlük boyutunu belirleyen kontrol parametresidir. SSASR metodunda  $C$  parametresi belirlenirken  $C$ 'nin boyutu spektral bant genişliğinin  $3/2$  katı alınmaktadır. Bundan dolayı SSASR’da sözlük oluşturulurken, her bir alt

sözlükten eşit sayıda atom seçilerek spektral bant genişliğinin 3/2 katı toplamda atom seçilir.

MSSASR’da ise C parametresi, Şekil 4.5(b)’de görüldüğü gibi sözlük atomu kullanım histogramı ilk atomlarda yoğunlaştığı için bant genişliğinin yarısı kadar seçilmek yeterli olmaktadır. Örneklenmiş veri matrisi, C’ye bağlı olarak seçilen öz vektörler ile Şekil 4.4’te gösterildiği gibi PCA uzayına taşınır ve bu sayede öz uzayda sözlük için atomlar oluşturulmuş olur.

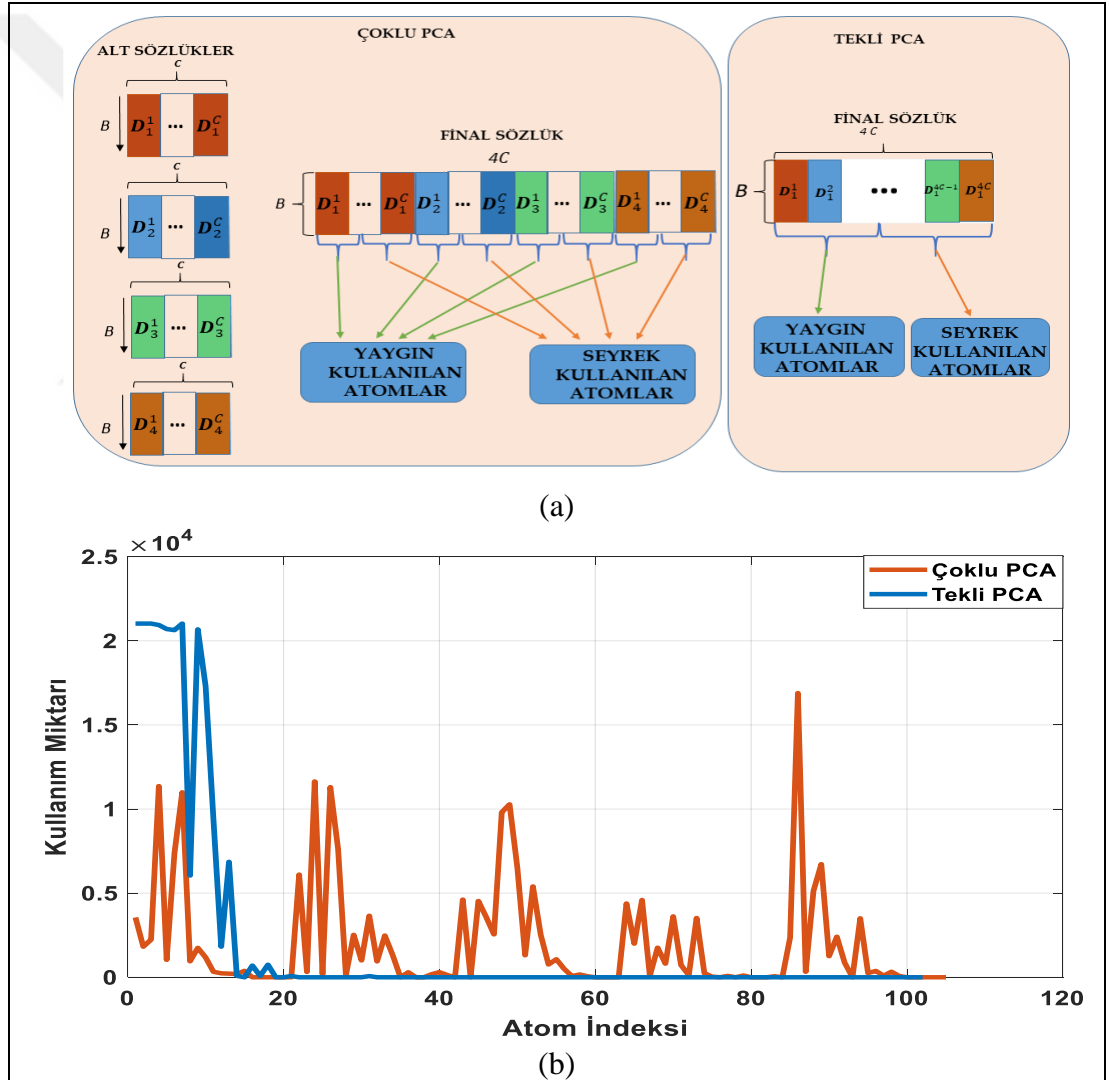


Şekil 4.4. PCA sonucu elde edilen katsayılar yardımıyla sözlüğün oluşturulması

Öz uzayda oluşturulan sözlük atomları, seyrek katsayı oluşturma aşamasında kullanılmaktadır. Sözlük atomlarının kullanım sayısı, sıkıştırma oranına göre değişim göstermektedir. 0,1, 0,2 bpp (bit per pixel) gibi düşük sıkıştırma oranlarında az sayıda sözlük atomuna ihtiyaç duyulurken 0,9, 1 bpp gibi yüksek sıkıştırma oranlarına çıktıkça ihtiyaç duyulan sözlük atomu sayısı artmaktadır. Bundan dolayı, her sıkıştırma oranı için sözlüğün hepsini alıcıya göndermeye ihtiyaç yoktur. Az kullanılan sözlük atomlarını atmak için sözlük atomlarının kullanım sayısı tespit edilerek, belirli bir kullanım oranının altında olan sözlük atomları atılmaktadır. Şekil 4.5(b)’de, MSSASR ve SSASR yöntemlerinde kullanılan, sırası ile tekli PCA ve çoklu PCA yaklaşımları ile elde edilen sözlük atomlarının kullanım histogramları verilmektedir. Çoklu PCA işlemini kullanan SSASR metodunda sözlük, Şekil 4.5(a)’da gösterildiği gibi alt sözlüklerin birleşmesinden oluşmaktadır. Alt sözlükleri oluşturmak için süperpiksellerden seçilen atomlar spektral açı uzaklığı (spectral angle distance, SAD) algoritmasıyla kümelenebilir [5]. Daha sonra her bir

kümeye PCA dönüşümü uygulanarak alt sözlükler oluşturulmaktadır. Oluşturulan her bir alt sözlüğün belirli sayıda atomu alınarak ana sözlük oluşturulmaktadır. Şekil 4.5(b)'de gösterildiği gibi bu yapıya sahip sözlükler, SAD sonucu bulunan küme merkezi sayısı kadar, sözlük atomlarının kullanım sıklığını farklı merkezlere dağıtmaktadır.

Oluşturulan sözlükten dolayı seyrek katsayıların enerjisi, küme merkezi sayısı kadar farklı merkeze dağılmaktadır. Şekil 4.6(a)'da ve Şekil 4.6(b)'de gösterildiği gibi seyrek katsayıların enerjisinin farklı merkezlere dağılması sonucu entropi kodlama verimsizleşmektedir.

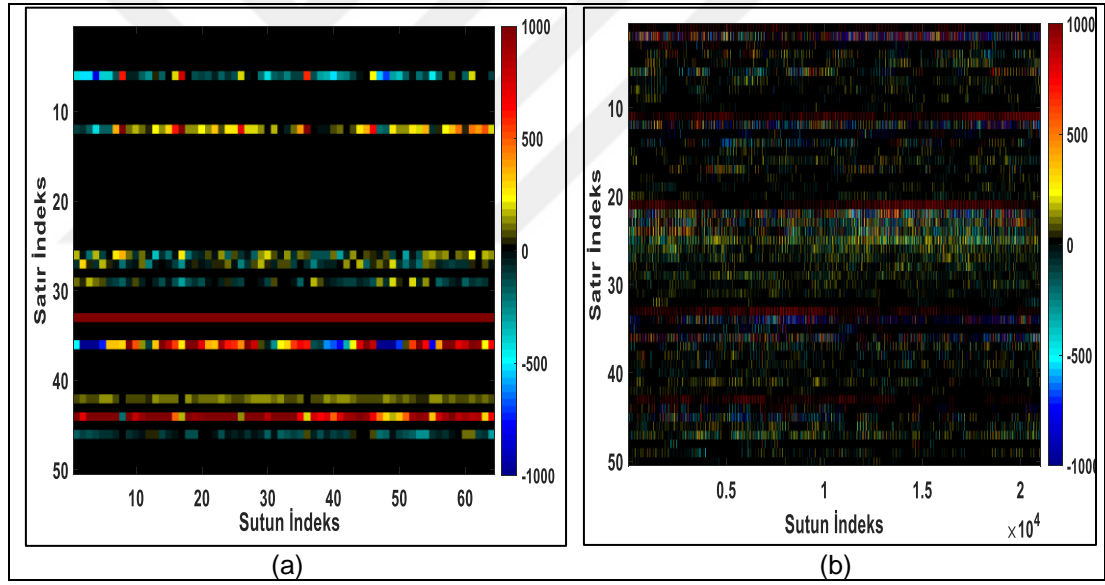


Şekil 4.5. (a) Çoklu ve tekli PCA için sözlük oluşturma (b) Çoklu ve tekli PCA'de kullanılan atomların histogramı.

Tekli PCA kullanıldığı zaman sözlükteki atomlar önem sırasına göre azalan bir yapıda sıralanmaktadır. Yüksek öneme sahip atomlar, seyrek katsayı oluşturmak için daha fazla kullanılırken, düşük öneme sahip atomlar Şekil 4.5(a)'da gösterildiği gibi daha az kullanılmaktadır. Böylece, seyrek katsayıların enerjisi de tek bir merkez etrafında toplanmaktadır.

#### 4.1.2.2.Sözlük oluşturmada tekli PCA' in katkısı

Seyrek katsayılar elde edilirken SSASR yönteminde SAD sonucu oluşturulan küme merkezi sayısına bağlı olarak seyrek katsayıların enerjisi dağılmaktadır. Şekil 4.6(a)'da tek bir süperpiksel için seyrek katsayı dağılım haritası görülmektedir. İlgili süperpikseli en iyi ifade edecek atomlar sözlükte dağınık bulunduğu için seyrek katsayılar farklı merkezlere dağılmıştır.

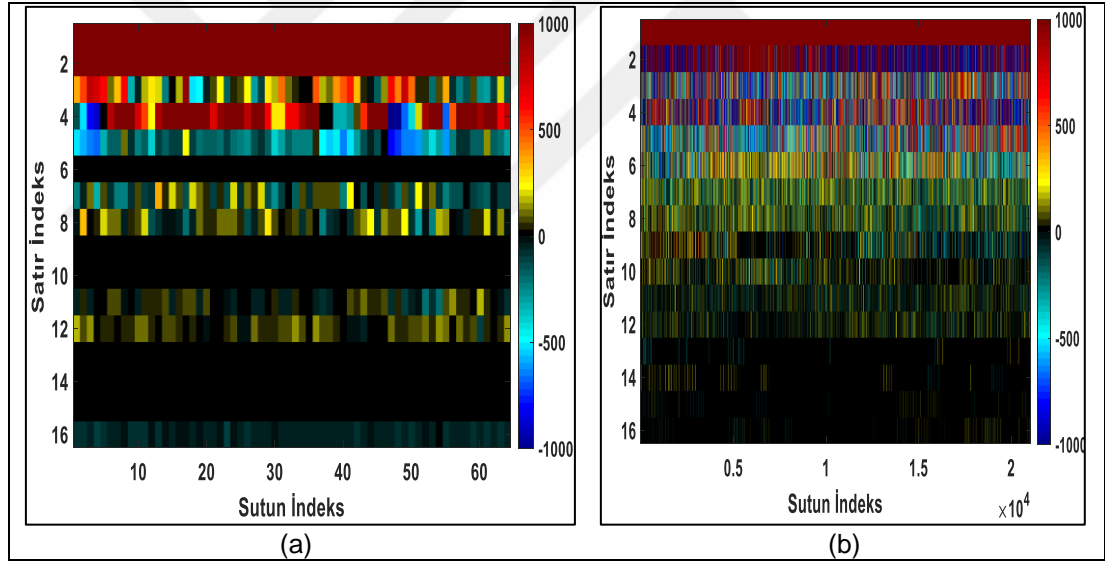


Şekil 4.6. (a) Tek bir süperpiksel için seyrek katsayı haritası SSASR (b) Bütün veri için sıralanmış seyrek katsayı haritası SSASR

Şekil 4.6(b)'de ise bütün katsayılar yan yana sıralı olarak dizildiğinde katsayıların önemli ölçüde dağıldığı görülmektedir. Şekil 4.6(a)'da ve Şekil 4.7(a)'da bulunan sütun indeks sayısı ilgili süperpikseldede bulunan atom sayısını göstermektedir ve her iki şekil içinde aynı süperpiksel kullanılmaktadır. Şekil 4.6(b)'de ve Şekil 4.7(b)'de bulunan sütun indeks sayısı ise kullanılan verideki toplam atom sayısını göstermektedir. Bu uygulamada Indian Pines verisi kullanıldığı için Şekil 4.6(b)'de

ve Şekil 4.7(b)'de bulunan c sütun indeks sayısı M ve N uzamsal boyutları ifade eden iki farklı değişken olmak üzere  $M \times N$  dir

Oluşturulan yeni yöntem sayesinde bütün süperpikselleri ifade edecek en iyi atomlar sözlüğün ilk sütunlarında toplandığı için seyrek katsayıları farklı merkezlere dağılmamaktadır. Şekil 4.7(a)'da ve Şekil 4.7(b) veriyi ifade edecek ilk 10 seyrek katsayı, sözlüğün ilk 16 sütununu kullandığı için seyrek katsayılar ilk 16 satırda toplu olarak bulunabilmiştir. Şekil 4.6(a) ve Şekil 4.6.b'de gösterildiği gibi aynı veri için SSASR, ilk 10 seyrek katsayıyı bulurken seyrek katsayıları 52 satıra dağıtmıştır. Uygulanan MSSASR sayesinde sözlük boyutu ilgili sıkıştırma oranı için  $(M \times 52)$ ' den  $M \times 16$ 'ya kadar düşürülmüştür. Buna bağlı olarak elde edilen seyrek katsayılarda  $(52 \times MN)$ ' den  $16 \times MN$  boyutuna kadar düşürülmüştür. Bu sayede, sıkıştırma için kodlanacak veri miktarı düşürülmüştür.



Şekil 4.7. (a) Tek bir süperpiksel için seyrek katsayı dağılım haritası MSSASR (b) Bütün veri için sıralanmış seyrek katsayı dağılım haritası MSSASR

#### 4.1.3. Seyrek katsayıların belirlenmesi

Seyrek tanımlama, sıkıştırılacak veriyi kullanılan sözlüğe bağlı olarak katsayılarının çoğu sıfırlardan oluşan matris şeklinde ifade etme işlemidir. Bu matris içerisinde bulunan katsayıların çoğunluğu, sıfır olduğu için bu verinin sıkıştırma işlemi verimli olmaktadır. Seyrek katsayıların bulunması işleminde dikgen eşleşen arama (Orthogonal Matching Pursuit, OMP), eş zamanlı dikgen eşleşen arama



(Simultaneous Orthogonal Matching Pursuit, SOMP), LASSO gibi birçok farklı yöntem kullanılmaktadır. Genellikle komşu olan ve benzer spektral imzaları paylaşan piksellerin eşzamanlı temsili seyrek tanımlama performansını arttırmaktadır. Süperpikseller ilintili pikselleri gruplandığı için eşzamanlı seyrek katsayı modeliyle tanımlanabilmektedir. MSSASR yöntemi de süperpiksel temelli olduğu için SOMP algoritmasını kullanmaktadır.

SOMP algoritması hem hız bakımından hem hesaplama kolaylığı açısından verimli olmaktadır. Seyrek katsayı hesabında verinin sıkıştırma kolaylığı açısından istenen en önemli faktör, seyrek katsayıların değerlerinin çoğunluğu sıfırlardan oluşmasıdır. Değerler sıfırlardan oluştuğu zaman veri iletim maliyetine çok küçük oranlarda yük getirmektedir.

SOMP algoritmasında karmaşık iç çarpım uzayında çalışılmaktadır. Sözlüğün her bir atomu  $\varphi_\omega$  ile gösterilmekte ve  $\omega$  atomun  $\Omega$  indeks setindeki indis numarasını belirtmektedir. Giriş sinyali olarak verilen süperpikseller  $Y_k$  ile gösterilmektedir ve  $B \times n_k$  boyutlarındadır. Verilen  $Y_k$  süperpikseline göre artık problem, eşzamanlı seyrek yaklaşım problemine (Simultaneous Sparse Approximation-SSA) dönmektedir [25].

$$\min_{X_k} \|Y_k - DX_k\|_F^2 \quad (4.1)$$

Denklem 4.1’de verilen  $X_k$  terimi k. süperpiksel için seyrek katsayı matrisini, D terimi ise sözlüğü ifade etmektedir. Sıfır olmayan satır sayısı (T) sıkıştırma oranına bağlı olarak değişmektedir. Her bir süperpiksel ayrı ayrı SOMP algoritmasına gönderilir ve her biri için ayrı ayrı T adet sıfır olmayan katsayı satırından oluşan seyrek katsayılar bulunur. Sıfır olmayan satır indislerini tutmak için  $\wedge_r$  parametresi kullanılır.

Bütün süperpikseller için Tablo 4.1’de verilen algoritma çalıştırılır ve seyrek katsayılar hesaplanır. Algoritmanın işleyiş şekli Tablo 4.1’de adım adım gösterilmektedir. Her bir süperpiksel için seyrek katsayılar ayrı ayrı hesaplanmaktadır.

Tablo 4.1. Uyarlamalı seyrek kodlama

---

Giriş Verileri: 1) Sözlük  $D$ , 2) İlgili süperpiksel  $Y_k$  3) Sıfır olmayan satır sayısı  $T$

Çıkış Verileri: 1) Sıfır olmayan satır indisleri  $\wedge_t$ , 2) Yaklaşım matrisi  $A_T$  3) Artıklık matrisi  $R_T$

---

- 1: Artıklık matrisine ilk değer olarak  $k$ . süperpiksel matrisini ata  $R_0 = Y_k$ ;
- 2: Sıfır olmayan indisleri tutan matrisi  $\wedge_0 = \phi$  olarak ata;
- 3: İterasyon sayısını  $t=1$  olarak ilklendir;
- 4: Döngü  $t = 1:T$
- 5: Aşağıdaki optimizasyon problemini çözen  $\lambda_t$  indisini bul ( $e_j$ :  $j$ . kanonik baz vektörü)

$$\max_{w \in \Omega} \sum_{j=1}^{n_k} \left| \langle R_{t-1} e_j, \varphi_w \rangle \right|$$

- 6: Maksimum sonucu veren indisi vektöre ekle  $\wedge_t = \wedge_{t-1} \cup \{\lambda_t\}$
- 7: Dikgen izdüşüm  $P_t$ 'yi,  $\wedge_t$ 'de indekslenen atomlar üzerinden hesapla
- 8: Yeni yaklaşım matrisini ve artıklığı hesapla  $A_t = P_t \times Y_k$ ,  $R_t = Y_k - A_t$ ;
- 9: Döngü sonu

---

#### 4.1.4. Seyrek katsayıları düzenleme işlemi

SSASR yönteminde görüntünün geri elde edilebilmesi için süperpiksel haritasına ihtiyaç duyulmaktadır. Bu nedenle, süperpiksel haritasının ayrıca entropi kodlanması gerekmektedir. Bu çalışma kapsamında geliştirilen yeni yöntem sayesinde süperpiksel haritasını kodlama ihtiyacı ortadan kaldırılmaktadır.

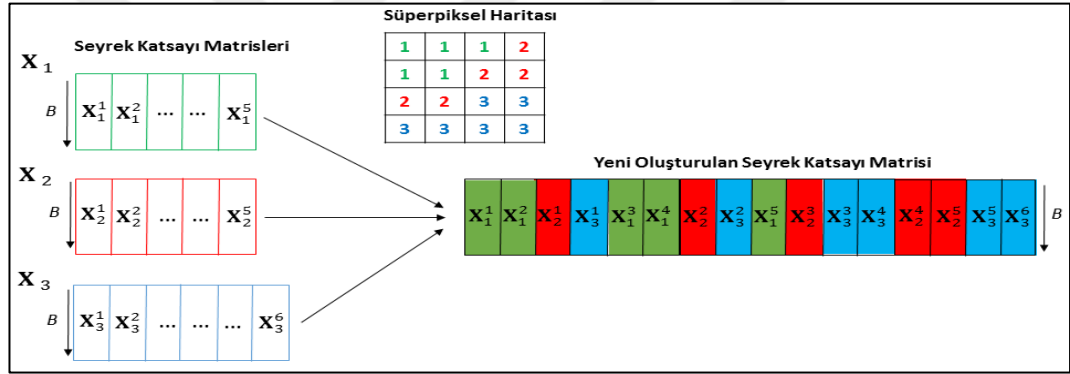
Süperpiksel haritasının kodlanma ihtiyacının ortadan kaldırılması büyük boyutlu verilerde ve süperpiksel sayısının yüksek olma durumunda karmaşık harita kodlanmasını ortadan kaldırdığı için yüksek süperpiksel sayılarına çıkıldıkça büyük avantaj sağlamaktadır. Bu yöntemin işlem aşamaları Şekil 4.8'de gösterilmiştir.

- Süperpiksel haritasında sütun boyunca yer alan piksellere sırası ile gidilmektedir. İlgili pikselin dahil olduğu süperpiksele ait seyrek katsayı

matrisindeki ilgili sütun, yeni oluşturulan seyrek katsayı matrisine Şekil 9’da verildiği gibi taşınmaktadır.

- Seyrek katsayı matrisi belli sıraya göre dizildiğinden, geri çatma sırasında süperpiksel haritasına gerek kalmadan, katsayıların sözlükle çarpılması yeterli olmaktadır.

SSASR yönteminde kullanılan çoklu PCA sonrasındaki seyrek katsayıların dağılımı bir bütünlük arz etmemektedir. MSSASR yönteminde önerilen tekli PCA ve sonrasında gerçekleştirilen katsayı yeniden sıralama sonrasında seyrek katsayılar ise yoğunlaşmış hal almaktadır. Şekil 4.6(a)’da ve Şekil 4.6(b)’de SSASR Şekil 4.7(a) ve Şekil 4.7(b)’de ise MSSASR yaklaşımları ile elde edilen seyrek katsayılar tek bir süperpiksel ve bütün görüntü için ayrı ayrı verilmiştir.



Şekil 4.8. Seyrek katsayı matrisi oluşturulurken izlenen yol

Şekilden görüldüğü üzere, MSSASR ile elde edilen seyrek katsayıların sahip olduğu bütünlük açıktır. Bu aşamada, sıkıştırma verimliliğinin artırılması için DPCM uygulanmaktadır. Bu amaçla, seyrek katsayı matrisinin yüksek genlikli (değerlikli) ilk R satırına DPCM uygulanmaktadır. Bu satırlar, matrisin enerjisinin %98’ini barındırmaktadır. Diğer satırlardaki katsayılar düşük olduğundan, DPCM uygulamak verimi düşürmektedir. Bu nedenle, sonraki satırlara ayrıca bir işlem uygulanmamaktadır. DPCM uygulamanın yöntemine katkısı Şekil 4.10’da gösterilmektedir.

#### **4.1.5. Nicemleme, entropi kodlama ve kod çözüme**

Bu kısımda, seyrek gösterimi elde edilen verinin sıkıştırılmasında tanımlı işlem aşamaları verilmektedir. İlk aşamada yapılan işlem, sıfır olmayan seyrek katsayı değerleri, sıfır olmayan seyrek katsayıların bulunduğu satır indisleri ve sözlüğün kodlanmasıdır. Entropi kodlama aşamasında Huffman kodlama kullanılmıştır [24]. Sıfır olmayan seyrek katsayılar ve sözlük değerleri tamsayı olmadığından, Huffman kodlama öncesinde nicemleme işlemi gerçekleştirilmektedir. Bu aşamada, tekdüze nicemleyici kullanılmıştır. Veriyi geri elde etme aşamasında Huffman kod çözücü kullanılmış ve elde edilen seyrek katsayılarla sözlük atomları çarpılıp geri elde etme işlemi tamamlanmıştır [24].

#### **4.1.6. Deneysel çalışmalar**

Bu bölümde, öncelikle kullanılan veri kümeleri hakkında bilgiler yer almaktadır. Ayrıca, önerilen yöntemle ait süperpiksel sayısının ve her süperpikselden seçilen örnek sayısının belirlenmesi için parametrik bir çalışma bulunmaktadır. İlâveten, MSSASR yöntemindeki geliştirmeleri analiz edebilmek amacıyla çoklu PCA ve DPCM yöntemlerinin SNR-Bit oranlarına katkıları paylaşılmaktadır. Son olarak, önerilen yöntemin performansı, farklı kalite metrikleri üzerinden literatürdeki SSASR [5], DWT-JPEG2000 [26], 3D-SPECK [27] ve 3D-TARP [28] yöntemlerle karşılaştırılmıştır.

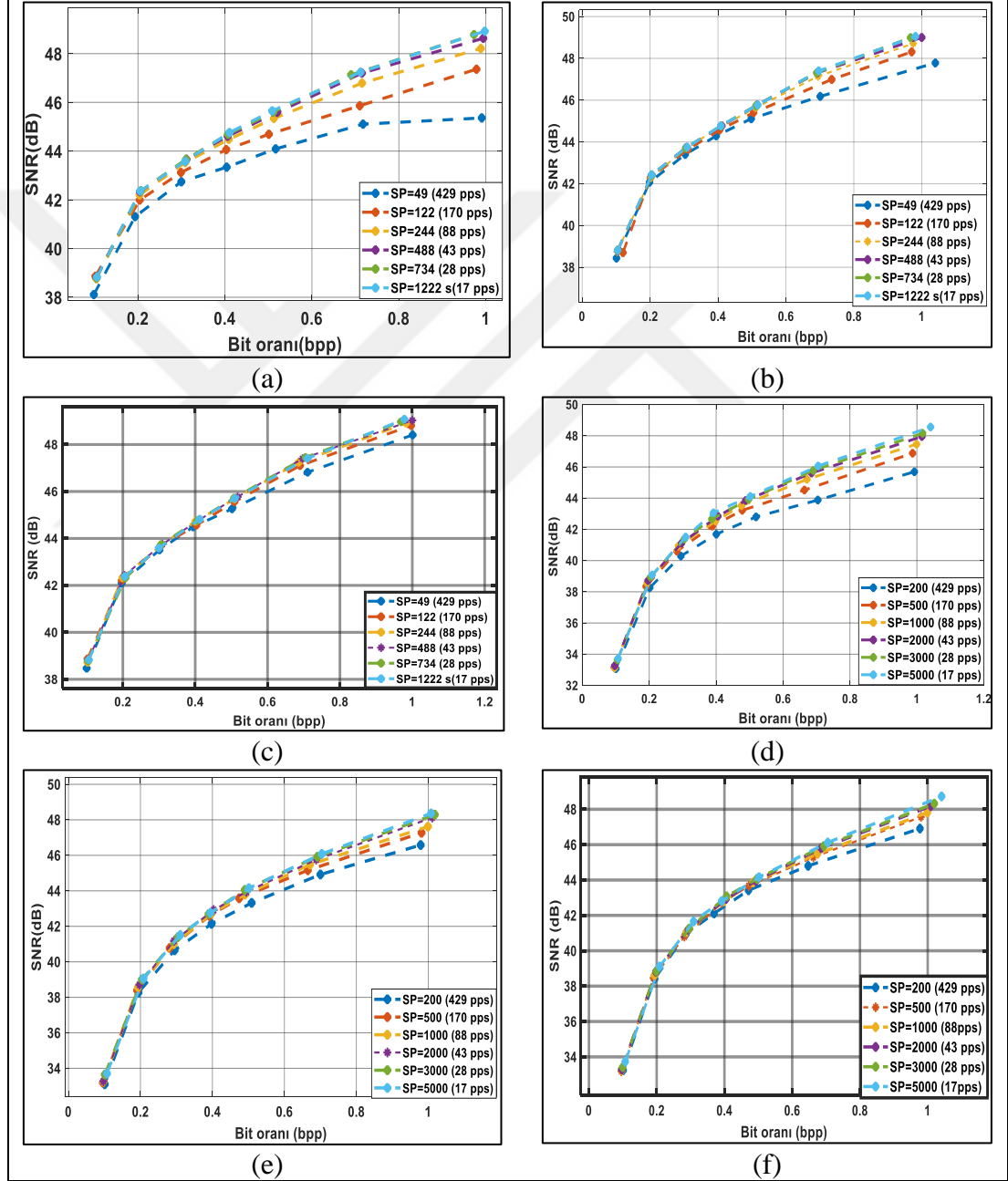
##### **4.1.6.1. Veri kümeleri**

Deneysel çalışmalar boyunca yöntemlerin performanslarının karşılaştırılması için Indian Pines, DC Mall, Moffett Field ve Jasper Ridge olmak üzere toplam dört farklı veri kullanılmıştır. Veri kümeleri hakkında detaylı bilgi Bölüm 1’de anlatılmaktadır.

##### **4.1.6.2. Süperpiksel sayısının belirlenmesi**

Veriyi benzer renkteki parçalara ayırarak seyrek katsayılarla ifade etmek için gerekli olan süperpiksel sayısının belirlenmesi problem oluşturmaktadır. Indian Pines için Şekil 4.9(a), Şekil 4.9(b), Şekil 4.9(c)’de Washington DC için Şekil 4.9(d), Şekil 4.9(e) ve Şekil 4.9(f)’de farklı süperpiksel değerlerinde ve farklı sayıda örnek

seçiminde SNR-bit oranı grafiği verilmiştir. verilmektedir. Ayrıca, sözlük oluşturmak için süperpiksellerden seçilecek örnek sayısı da bilinmemektedir. Grafikler incelendiğinde her bir veri için gerekli olan süperpiksel sayısı, veri boyutuna bağlı olduğu görülmektedir. Veri boyutu büyüdükçe gerekli olan süperpiksel sayısı artmaktadır. Süperpiksel başına düşen ortalama piksel sayısı 43 pps (pixel per superpixel) olduğu zaman, veriyi ifade etmek için yeterli olduğu görülmektedir.

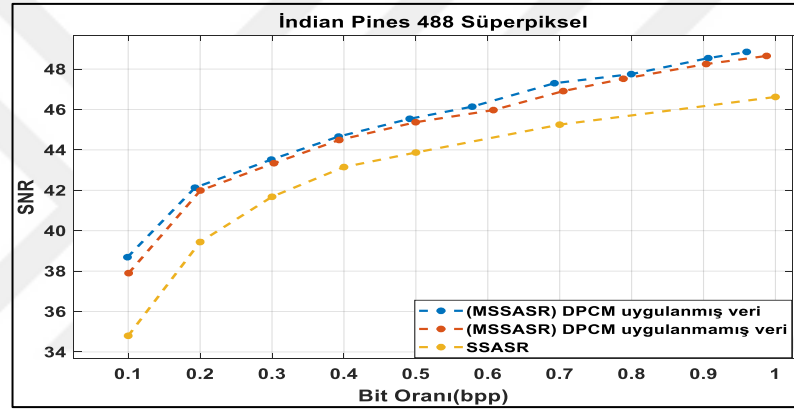


Şekil 4.9. Farklı süperpiksel sayısı ve örnek seçimi için SNR-Bit grafikleri

Sözlük oluşturmada fazla işlem yükünden kaçmak için her bir süperpikselden 3 örnek almak yeterli olmaktadır. Süperpiksel başına düşen ortalama piksel sayısı 43pps olması için Indian Pines verisi 488 süperpiksele, Washington DC verisi ise 2000 süperpiksele bölünmesi yeterli olmaktadır.

#### 4.1.6.3. Tekli PCA ve DPCM kullanılması başarımına katkısı

Şekil 4.10’da tekli ve çoklu PCA işleminin yöntem sonuçlarına etkisi incelenmektedir. Buna göre, tekli PCA işlemi, çoklu PCA’ye göre yaklaşık 2.5 dB’lik bir kazanç sağlamaktadır. Bu da tekli PCA kullanan MSSASR yönteminin oldukça verimli hale getirildiğini göstermektedir.



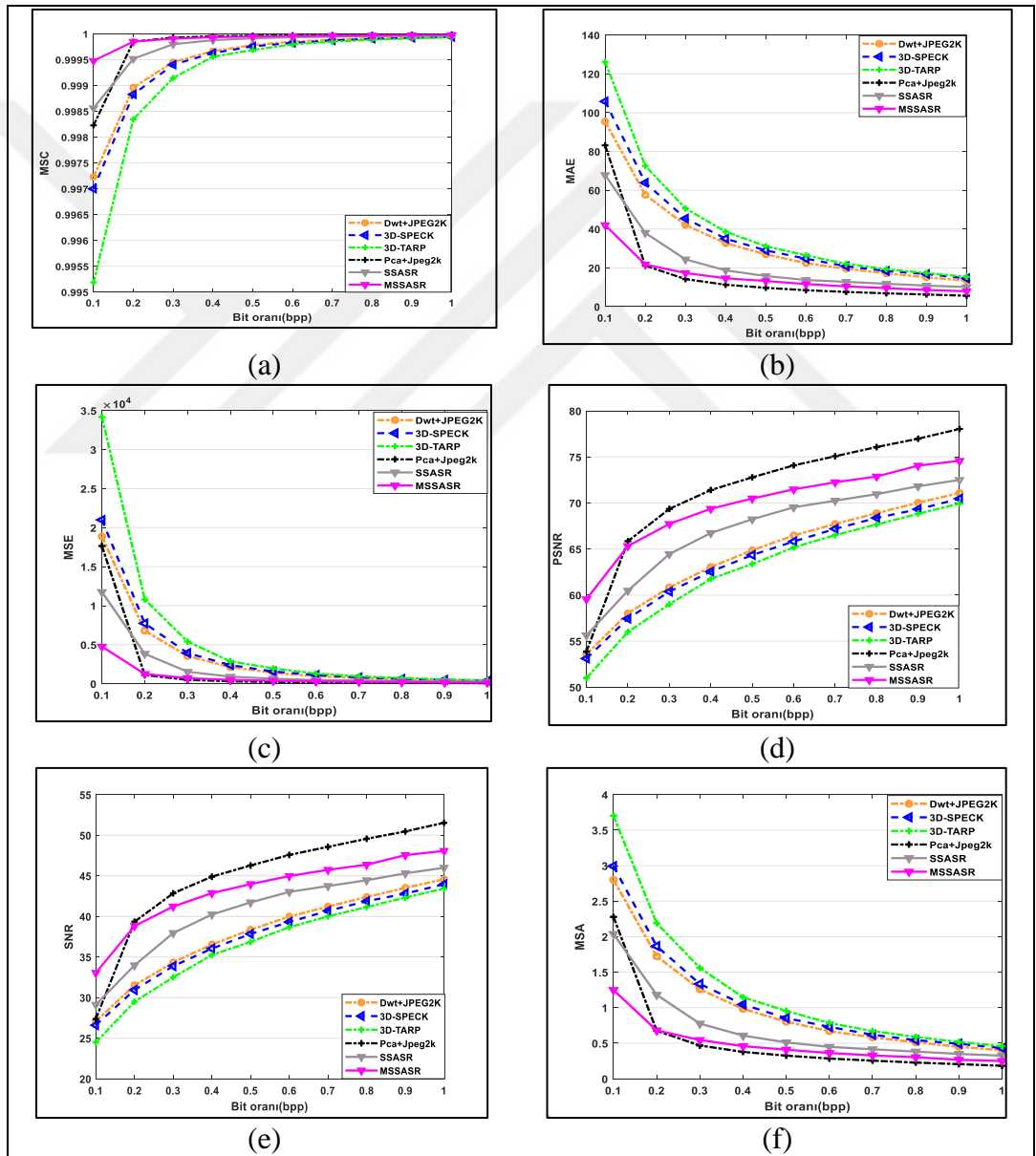
Şekil 4.10. Tekli PCA ve DPCM işleminin yöntem katkısı

Ayrıca, Şekil 4.10’da DPCM işlemi varken ve yokken MSSASR yöntemiyle elde edilen sonuçlar paylaşılmıştır. Buna göre, DPCM’in katkısı tekli PCA’in katkısına göre sınırlı olmaktadır. DPCM ilk R adet satıra uygulandığı için veriye sabit bir sıkıştırma katkısı sağlamaktadır. Bu katkı, en fazla düşük bitlerde görülmektedir.

#### 4.1.6.4. Sıkıştırma performansının güncel yöntemlerle karşılaştırılması

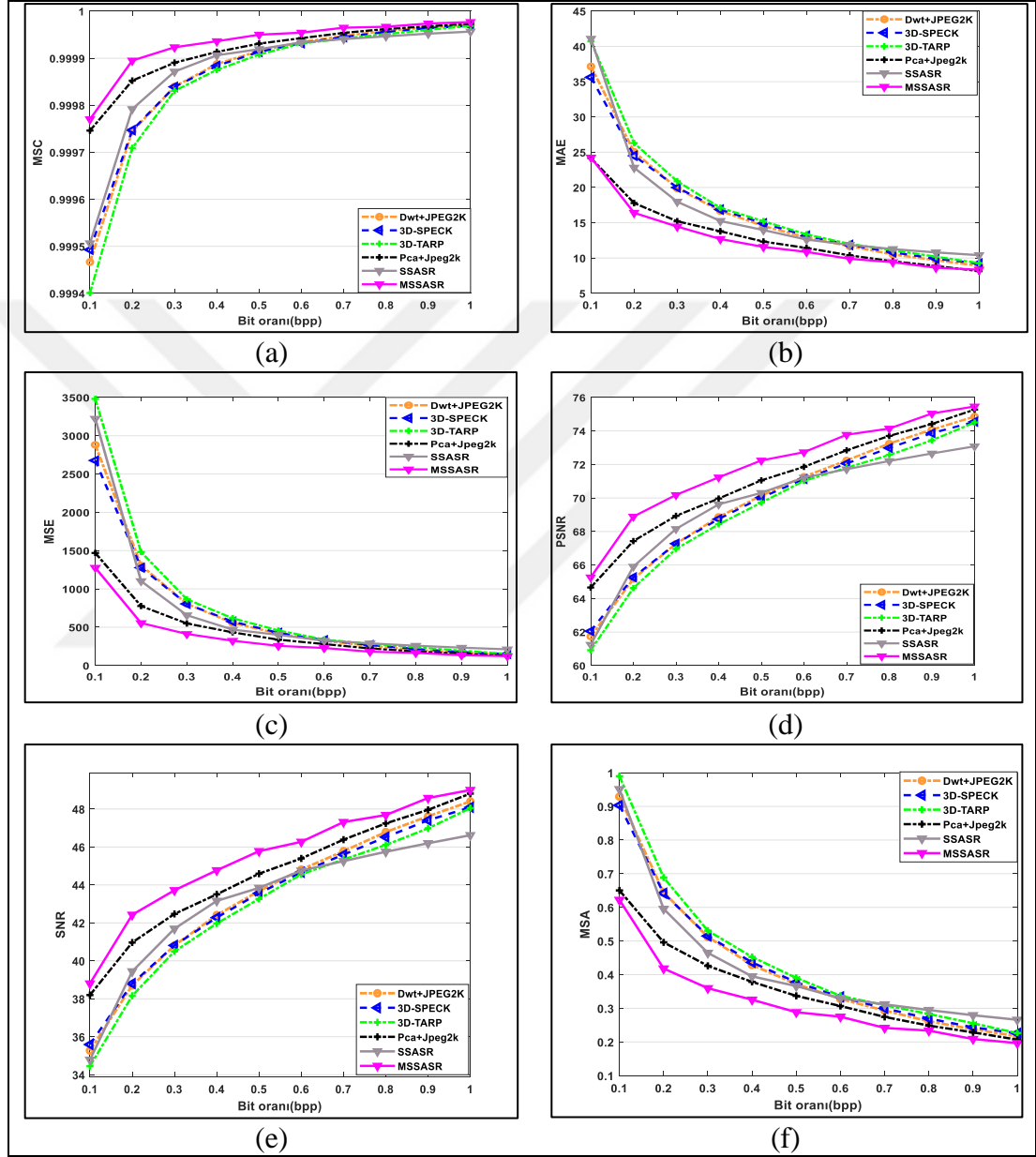
Önerilen MSSASR yöntemi ve DWT-JP2K, 3D-SPECK, 3D-TARP, SSASR yöntemlerinin farklı metriklerdeki bit oranına bağlı grafikleri Washington DC için Şekil 4.11(a), Şekil 4.11(b), Şekil 4.11(c), Şekil 4.11(d), Şekil 4.11(e) ve Şekil 4.11(f)’de verilmektedir. Indian Pines için Şekil 4.12(a), Şekil 4.12(b), Şekil 4.12(c), Şekil 4.12(d), Şekil 4.12(e) ve Şekil 4.12(f)’de verilmektedir. Jasper Ridge için Şekil 4.13(a), Şekil 4.13(b), Şekil 4.13(c), Şekil 4.13(d), Şekil 4.13(e) ve Şekil 4.13(f)’de

verilmektedir. Moffett Field için Şekil 4.14(a), Şekil 4.14(b), Şekil 4.14(c), Şekil 4.14(d), Şekil 4.14(e) ve Şekil 4.14(f)'de verilmektedir. Sıkıştırma sonrasında geri elde edilen verinin genel kalitesini/bozunumunu en iyi özetleyen metrikler SNR, MSA ve MSC metrikleridir. Grafiklerin tamamında MSSASR yönteminin diğer tüm yöntemlerden daha iyi bir performans sergilediği görülmektedir. SNR, PSNR ve MSC metriklerinde orijinal veriye yaklaşmak için istenen sonuçların yüksek olmasıdır. MSA, MAE, MSE metriklerinde ise ne kadar düşük ise orijinal veriye o kadar yakın olmaktadır.



Şekil 4.11. Washington DC için metrik sonuçları

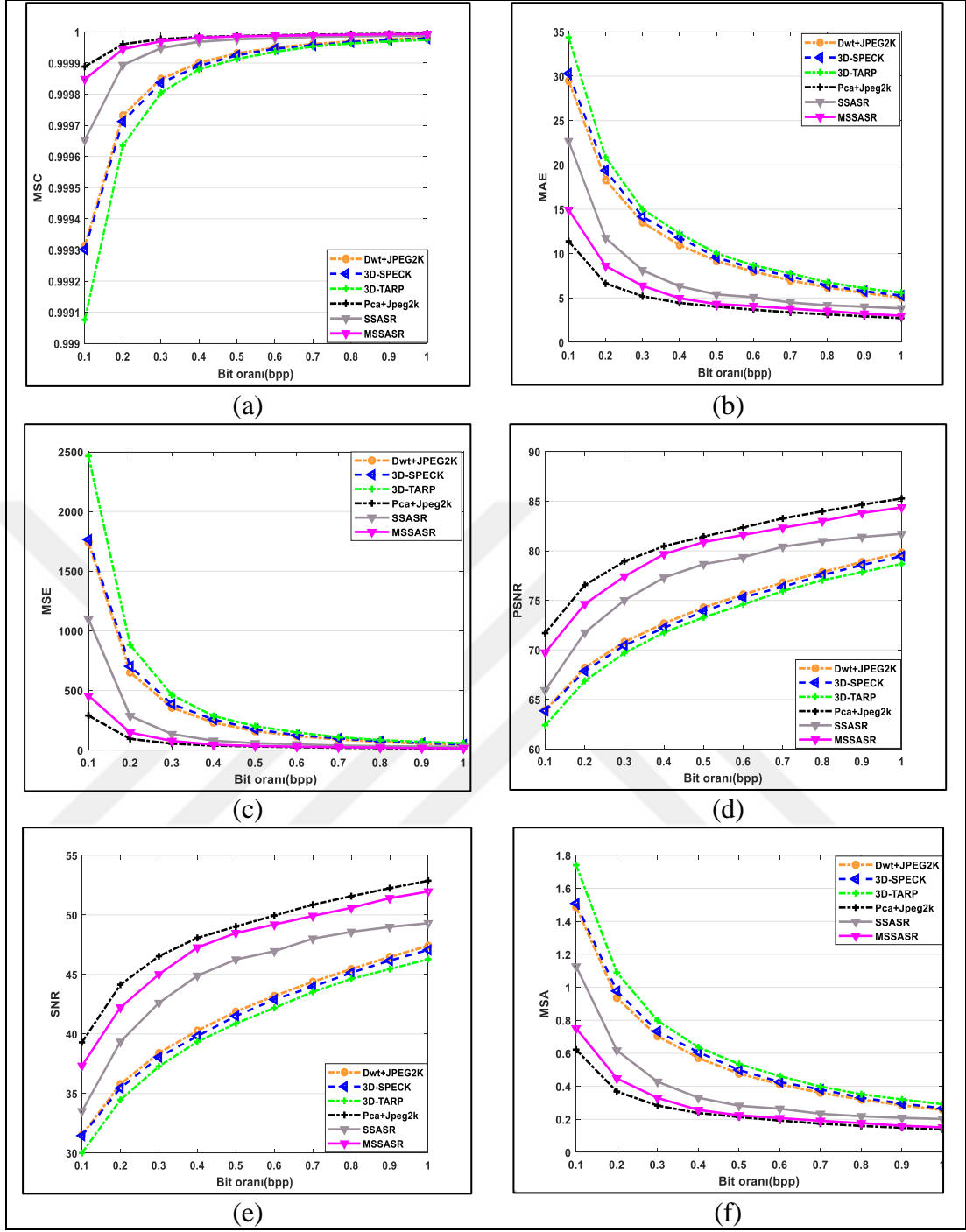
Şekil 4.11’de MSSASR yönteminin Washington DC için metrik sonuçları verilmiştir. PCA+JPEG2K yöntemi hariç diğer yöntemler karşısında tüm parametreler için önde olmaktadır. PCA+JPEG2K yöntemi karşısında ise düşük bit oranlarında önde olmaktadır.



Şekil 4.12. Indian Pines için metrik sonuçları

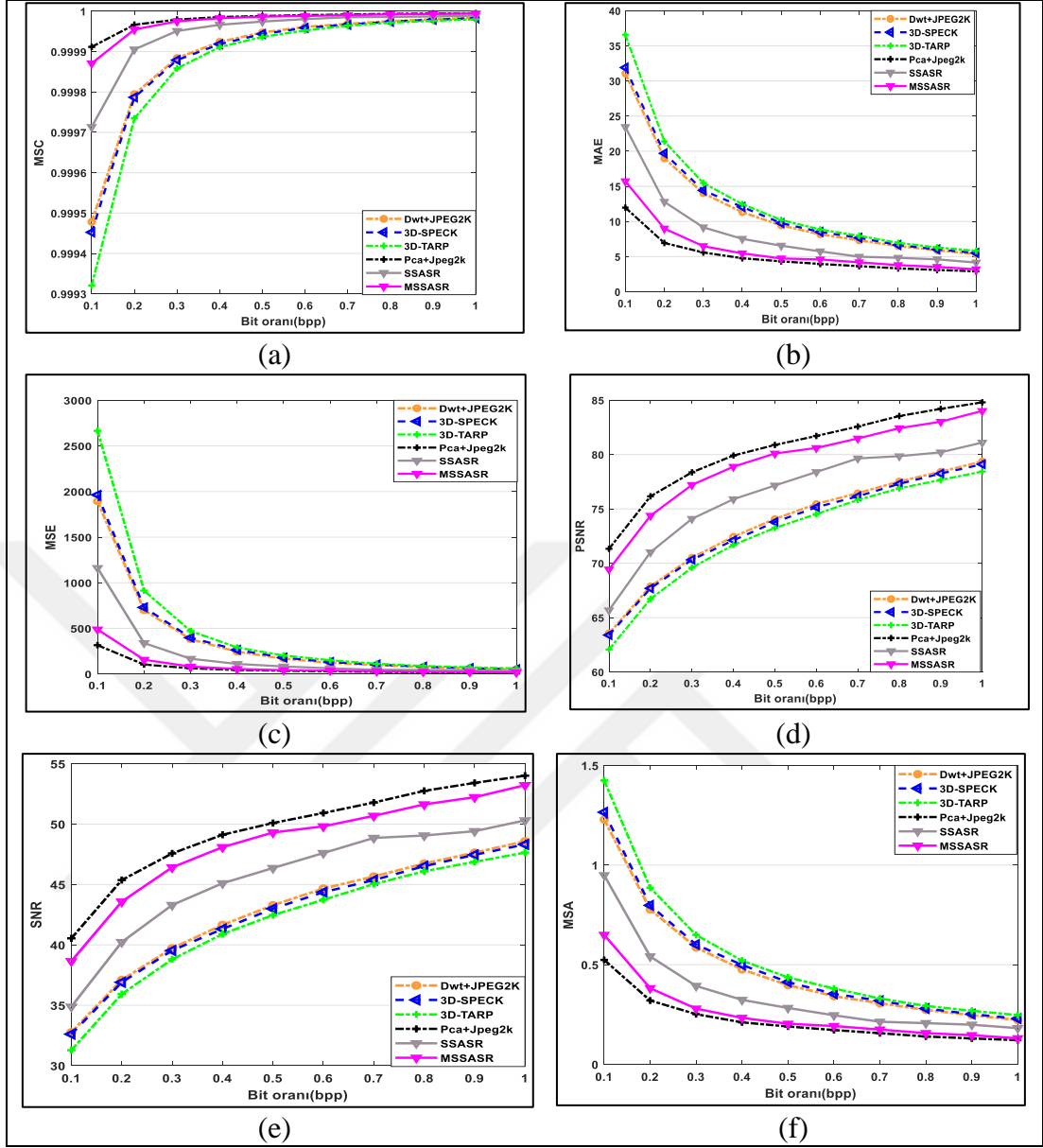
Şekil 4.12’de MSSASR yönteminin Indian Pines için metrik sonuçları verilmiştir. MSSASR yöntemi, karşılaştırmada verilen yöntemler karşısında tüm parametreler için önde olmaktadır.





Şekil 4.13. Jasper Ridge için metrik sonuçları

Şekil 4.13'te MSSASR yönteminin Jasper Ridge için metrik sonuçları verilmiştir. PCA+JPEG2K yöntemi hariç diğer yöntemler karşısında tüm parametreler için önde olmaktadır. PCA+JPEG2K yöntemi karşısında ise düşük bit oranlarında önde olmaktadır.



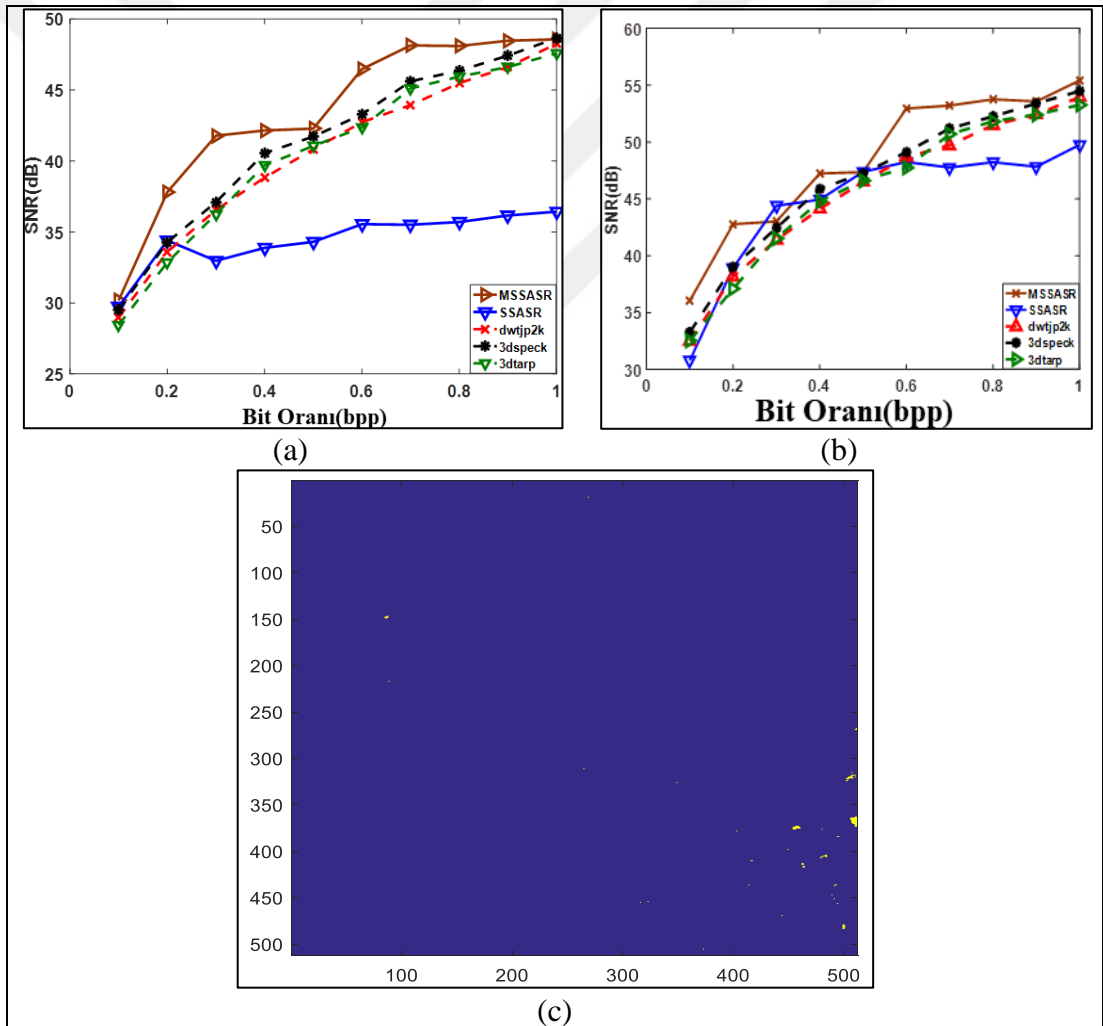
Şekil 4.14. Moffett Field için metrik sonuçları

Şekil 4.14'te MSSASR yönteminin Moffett Field için metrik sonuçları verilmiştir. PCA+JPEG2K yöntemi hariç diğer yöntemler karşısında tüm parametreler için önde olmaktadır. PCA+JPEG2K yöntemi karşısında ise düşük bit oranlarında önde olmaktadır.

#### 4.1.6.5. Anomali koruma

Hiperspektral görüntülerde yukarıda kullanılan kalite metrikleri tüm veri üzerinde çalıştırıldığından verinin genel bozunumu hakkında bilgi sağlamaktadır. Ancak,

hiperspektral görüntü içerisinde verinin genelinden daha farklı davranışa sahip spektral imzalar da bulunmaktadır. Bu imzaların bulunduğu pikseller anomali pikseller olarak tanımlanmaktadır. Önerilen yöntemin anomali pikselleri koruyup korumadığını bulmak için Moffett Field ve Jasper Ridge verilerine RX anomali tespiti yöntemi uygulanarak anomali haritası elde edilmektedir. Şekil 4.15(c)'de Moffett için anomali haritası görülmektedir. Bu haritadan, anomali değeri en yüksek 150 piksel her iki veri için seçilmektedir. Seçilen pikseller sarı renk ile gösterilmektedir. Diğer pikseller ise mavi renk ile görünmektedirler. Resmin boyutu 512 olduğu için aralık bu boyuta kadar uzamaktadır. Seçilen anomali pikseller SNR hesabı kıyaslamasında kullanılır.

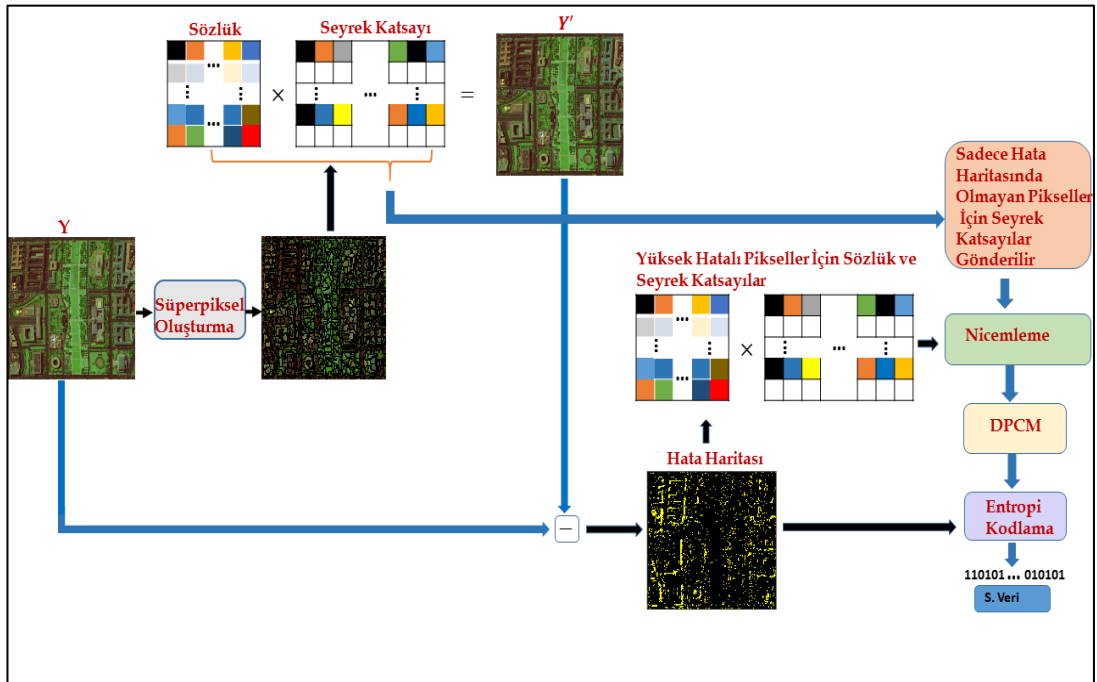


Şekil 4.15. (a) Jasper Ridge verisi için ortalama SNR grafiği (b) Moffett Field verisi için ortalama SNR grafiği (c) Moffett için anomali haritası

Moffett Field ve Jasper Ridge verileri farklı yöntemlere göre sıkıştırılıp iletikten sonra her bir yöntem için daha önce tespit edilen anomali pikseller üzerinden ortalama SNR deęerleri Şekil 4.15(a) ve Şekil 4.15(b)'de gösterilmektedir. Grafikler incelendiğinde her iki veride de MSSASR yönteminin anomali piksellerini dięer yöntemlere göre daha iyi koruduęu görülmektedir.

#### 4.2. Hata düzeltmeli süperpiksel temelli seyrek gösterimle sıkıştırma yöntemi (ECSSASR)

SSASR yöntemi bazı pikselleri geri elde aşamasında aşırı bozduęu için yaygın kullanılan birçok metrikte literatürde kullanılan birçok yõteme göre kötü sonuç vermektedir. SSASR yönteminin sahip olduęu özellikler arasında, bozunuma uğrayan pikselleri tespit edip düzeltme işlemi uygulayan bir yapı mevcut deęildir. Bu sebeple bozunuma uğrayan pikseller fazlaca olmaktadır. Geri elde etme aşamasından önce, bozunuma uğrayan pikseller tespit edilip düzeltilemezse, başarımlı tespiti için kullanılan metrik sonuçlarında yöntem kötü sonuç vermektedir. Bu tez çalışması kapsamında, SSASR'nin mevcut sorununu gidermek için bozunuma uğrayan pikselleri kodlarken ayrı bir sözlük ve seyrek katsayı hesabı yapılmaktadır. Önerilen yöntemin işleyiş şeması Şekil 4.16'da verilmektedir.



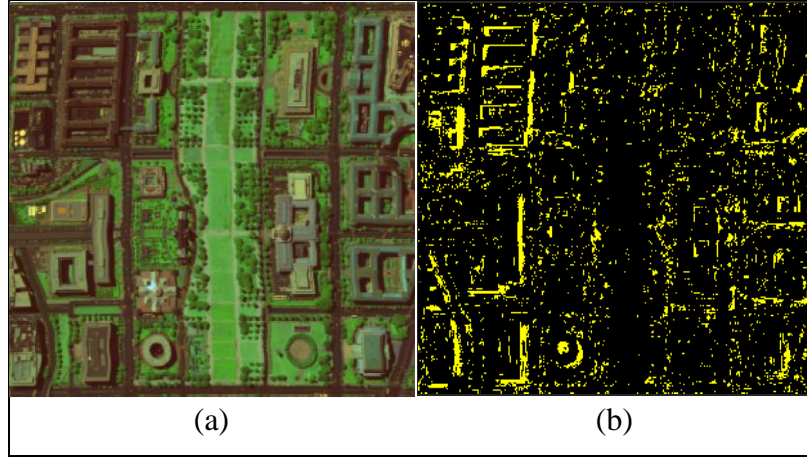
Şekil 4.16. Yöntemin genel işleyişi

Şekil 4.16'da verilen yöntemin işleyiş adımları aşağıda verilmektedir. Verilen adımlar sırasıyla ilerleyen aşamalarda anlatılacaktır.

- Süperpiksel haritasının, spektral benzerlik ölçüsüyle oluşturulması. Süperpiksel haritasının oluşturma aşamaları, Bölüm 3'te verilmiştir. SSASR ile aynı metodu kullanmaktadır.
- Süperpiksel haritasındaki her bir süperpikselden rastgele örnekleme yapılarak spektral matris oluşturulması.
- Spektral matrise PCA uygulanarak sözlüğün oluşturulması.
- SOMP algoritması ile süperpikselleri ifade edecek seyrek katsayı hesabının yapılması.
- SNR dağılım haritası hesaplanarak hata haritasının oluşturulması ve hata haritasındaki pikseller için sözlük ve seyrek katsayıların oluşturulması.
- Son olarak hata haritası, yüksek hatalı pikseller için sözlük ve seyrek katsayılar, hata haritasında olmayan pikseller için sözlük ve seyrek katsayılar sıkıştırılıp iletilmektedir.

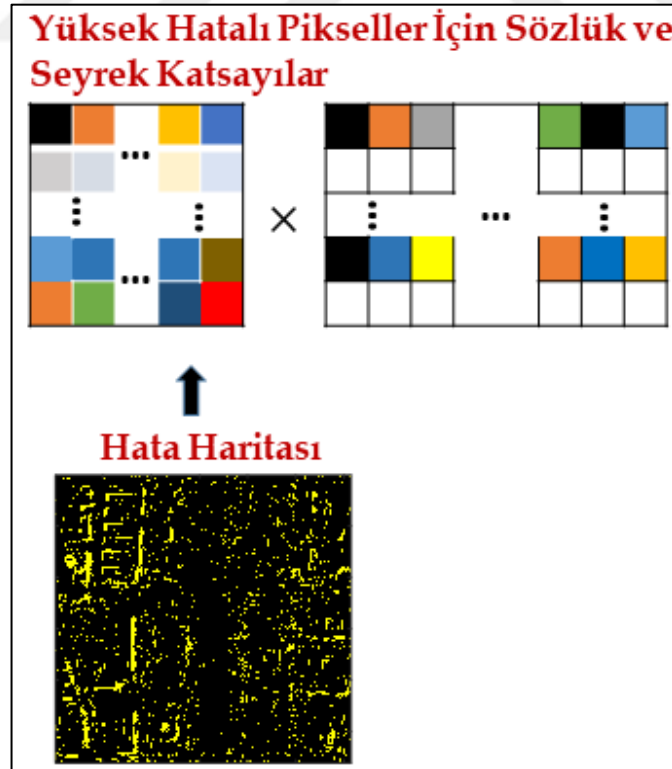
Yöntemin ilk aşamasında, giriş görüntüsü, spektral benzerlik ölçüsüne dayanarak, süperpikselle ayrılmaktadır. Oluşturulan süperpikseller, spektral benzerlik oranı yüksek pikselleri barındırmaktadır. Süperpiksel sayısının belirlenmesi işlemi, Bölüm 3'deki seçim işlemine göre yapılmaktadır. Oluşturulan süperpiksellerden rastgele elemanlar seçilerek, sıralanmaktadır. Sıralaması yapılan elemanlar SAD yardımıyla uzaklık hesabına göre farklı farklı merkezlere atanmaktadır. Şekil 3.3'de atama işlemleri gösterilmektedir.

Örneklenecek kümelenmiş veriler PCA dönüşümüne sokularak sözlük için atomlar öz uzayda oluşturulmaktadır. Şekil 3.4'de sözlük için oluşturulan atomlar görülmektedir. Fakat bu atomların hepsi sözlükte kullanılmamaktadır. Şekil 3.4'de görüldüğü gibi ilk C adet atom nihai sözlükte kullanılmaktadır. Oluşturulan sözlükle beraber süperpikseller ayrı ayrı SOMP algoritmasına sokularak seyrek katsayılar hesaplanmaktadır. Seyrek katsayı hesabında Bölüm 3.3'te anlatılan SOMP algoritması kullanılmaktadır. Oluşturulan sözlük ve katsayılar yardımıyla Şekil 4.17'de gösterilen hata haritası hesaplanmaktadır.



Şekil 4.17. Kullanılan veri (a) Washington DC ve (b) hata haritası

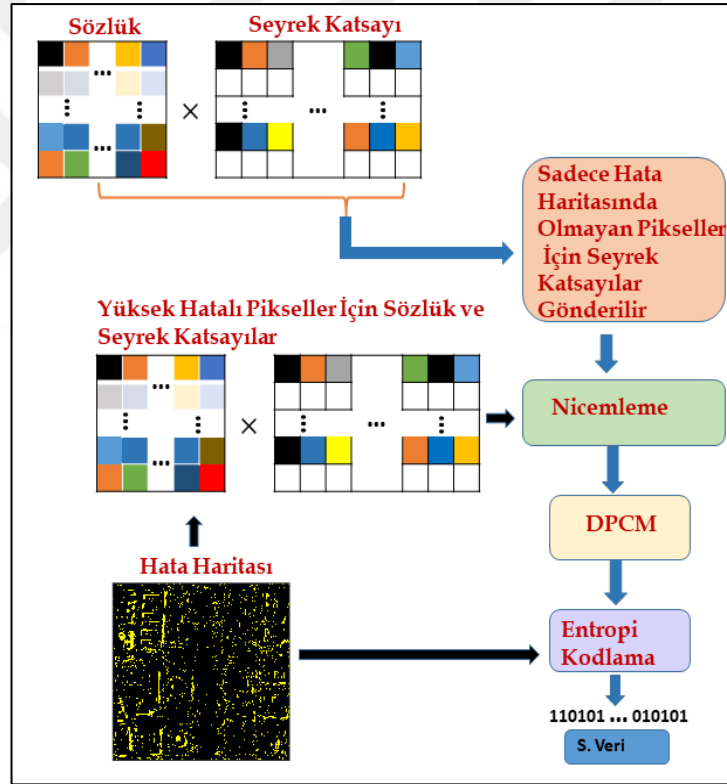
Hata haritası elde edildikten sonra hata miktarı yüksek olan pikseller için yeni bir sözlük ve seyrek katsayı matrisi oluşturulmaktadır. Yeni bir sözlük ve katsayı matrisi oluşturmaktaki amaç, kötü ifade edilen pikselleri iyi ifade edilebilir bir tanımlama elde etmektir. Şekil 4.18’de, hatalı pikseller için sözlük ve katsayı oluşturma gösterilmektedir.



Şekil 4.18. Hatalı pikseller için oluşturulan sözlük ve katsayılar

Hatalı pikseller için oluşturulan sözlük aşamaları, Bölüm 3.2’de anlatılan işlem basamaklarıyla aynı olduğu için burada tekrar açıklanmamıştır. Ayrıca hatalı pikseller için oluşturulan seyrek katsayılar Bölüm 3.3’te anlatılan SOMP algoritması yardımıyla oluşturulmuştur.

Son aşamada sıkıştırma işlemi için hata haritası, yüksek hatalı pikseller için sözlük ve seyrek katsayılar ve yüksek hata içermeyen pikseller için oluşturulan sözlük ile katsayılar iletilmektedir. Hata haritası hariç sıkıştırma için gönderilen diğer veriler ilk olarak nicemleme işlemine tabi tutulmaktadır. Daha sonra bu verilere sırasıyla DPCM ve entropi kodlama uygulanmaktadır. Şekil 4.19’da işlem basamakları gösterilmektedir. Hata haritası ise sıkıştırılmadan iletilmektedir.

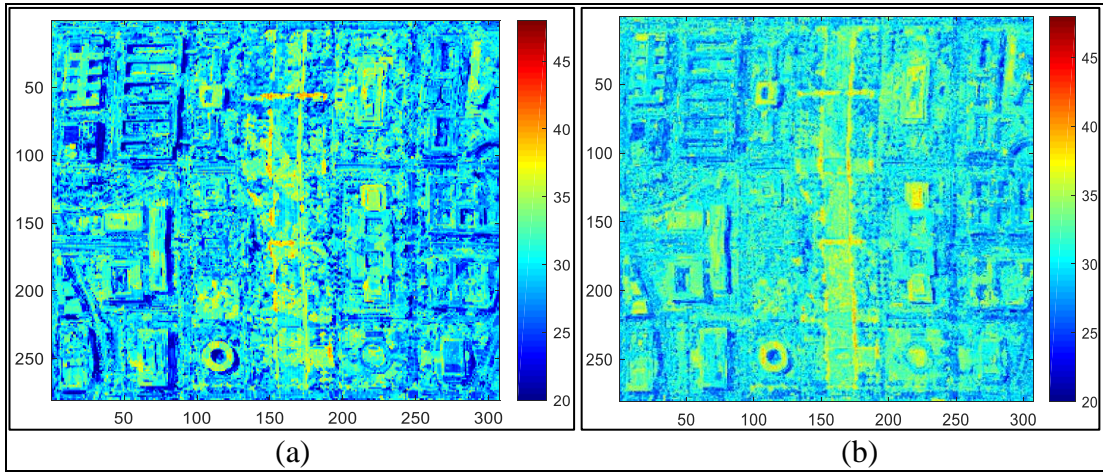


Şekil 4.19. Sıkıştırma aşamaları

#### 4.2.1. Deneysel çalışmalar

Yöntemlerin karşılaştırılması aşamasında Hydice sensörü ile alınan DC Mall görüntüsü kullanılmıştır. Görüntünün özellikleri Bölüm 1.1’de verilmektedir. Süperpiksel temelli sözlük öğrenmesi ve seyrek katsayı belirlenmesine dayanan

sıkıştırma yöntemlerinde süperpiksel sayısı, sözlük öğrenmesi için her bir süperpikselden alınan örnek sayısı ve sözlük boyutu gibi parametreler önem taşımaktadır [5]. Süperpiksel sayısı az olduğu zaman veride bulunan her bir farklı bölgeyi ifade edecek bir süperpiksel olmayacaktır. Veriyi ifade edecek yeterli sayıda süperpiksel olmadığı zaman her bir süperpiksel içerisinde o bölgeye uymayan karakteristik içeren pikseller bulunma ihtimali artmaktadır. Seyrek katsayıları bulmak için kullanılan SOMP algoritmasına göre herhangi bir süperpiksel için seyrek katsayı belirleme işlemi sırasında süperpikselin hatasını minimize edecek sözlüğün ilgili atomu seçilmektedir. Seçilen bu atom süperpiksel içerisindeki verinin geneline bağlı olarak belirlendiği için süperpiksel içinde bulunan farklı bölgeye ait piksellerin seyrek katsayıları iyi bir şekilde ifade edilemeyecektir. Süperpiksel sayısı olarak Washington DC verisi için  $K=2000$  olarak belirlenmiştir. Her bir süperpikselden seçilen örnek sayısı ilgili süperpikseli ifade etmek için yeterli sayıda olmalıdır. Ayrıca seçilen bu örnekler K-Ortalamlar yöntemiyle gruplandıktan sonra PCA işlemine tabi tutulmaktadır çalışma performansını azaltmamak için seçilen örnek sayısı mümkün olduğu kadar az olması ideale en uygun olanı olmaktadır. Bu işlem sırasında her bir süperpikselden üç örnek alınmıştır.

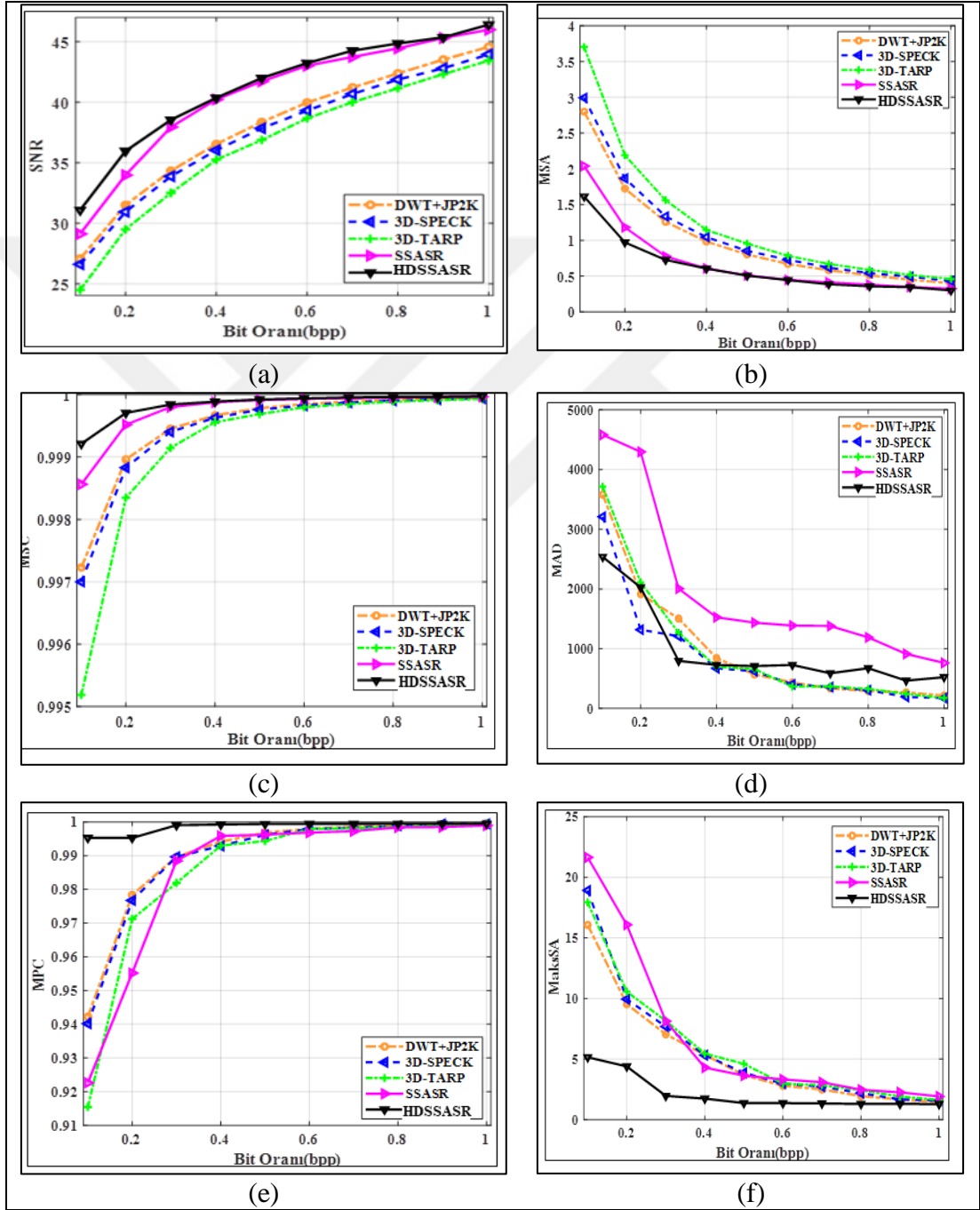


Şekil 4.20. (a) SSASR SNR dağılımı. (b) ECSASR SNR dağılım

K-Ortalamlar uygulanırken başlangıç küme merkezleri ve küme merkezi sayısı belirlenirken SAD algoritması kullanılmıştır ve bütün piksellerle arasındaki açı  $25^\circ$ 'den büyük olan pikseller küme merkezi olarak belirlenmiştir. Sözlük boyutu ilk aşamada verinin bant genişliğinin yarısı olarak belirlenmektedir ardından az



kullanılan atomlar atılarak sözlük boyutu güncellenmektedir. Şekil 4.20(b)'de Washington DC verisinin 0,1 bit sıkıştırma oranındaki SNR dağılım haritasına bağlı olarak, hatanın en yüksek olduğu 8000 piksel sarı ile ifade edilmektedir. Bu sayı toplam piksel sayısının %10'una denk gelmektedir. Şekil 4.20(a)'da SSASR yönteminin SNR dağılım haritası verilmektedir.



Şekil 4.21. Yöntemlere ait metrik sonuçları

SSASR için verilen SNR dağılım haritasında bazı piksellerin SNR değeri 8 dB'ye kadar düştüğü görülmektedir. Bu durum, SSASR yönteminin ilgili pikselleri geri elde ederken yüksek oranda bozduğunu ifade etmektedir. Uygulanan ECSSASR sayesinde Şekil 4.20(b)'de gösterilen SNR dağılım haritasına bakıldığı zaman Şekil 4.20(a)'da mavi renk ile ifade edilen düşük SNR değerine sahip piksellerin orta ve yüksek seviye renk aralığına çekildiği görülmektedir ayrıca en düşük SNR'a sahip pikselinde 22 dB SNR çıkarıldığı görülmektedir.

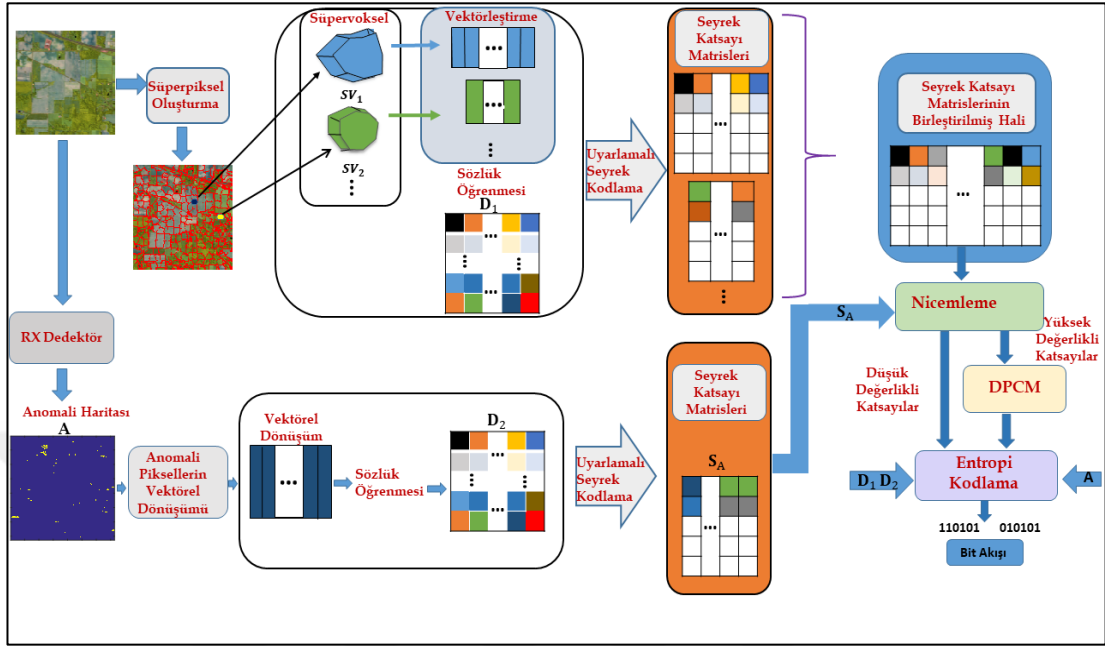
Önerilen yöntemin performansının karşılaştırılması için sinyal gürültü oranı (SNR), ortalama spektral açı benzerliği (MSA), ortalama spektral ilinti (MSC), maksimum spektral açı (MaksSA), maksimum mutlak uzaklık (MAD), minimum Pearson katsayısı (MPC) gibi kalite ve benzerlik metrikleri kullanılmıştır.

Önerilen ECSSASR yöntemi ve DWT-JP2K, 3D-SPECK, 3D-TARP, SSASR yöntemlerinin farklı metriklerdeki bit oranına bağlı grafikleri Washington DC için Şekil 4.21(a), Şekil 4.21(b), Şekil 4.21(c), Şekil 4.21(d), Şekil 4.21(e) ve Şekil 4.21(f)'de verilmektedir. Sıkıştırma sonrasında geri elde edilen verinin kalitesini/bozunumunu en iyi özetleyen metrikler SNR, MSA ve MSC metrikleridir. Şekil 4.21'de ECSSASR yönteminin diğer tüm yöntemlerden daha iyi bir performans sergilediği görülmektedir. SNR ve MSC metriklerinde orijinal veriye yakınlaşmak için istenen sonuçların yüksek olmasıdır. MSA'da ise ne kadar düşük ise orijinal veriye o kadar yakın olmaktadır. MaksSA, MPC, MAD metrikleri ise en kötü pikselin ne kadar bozulduğunu ifade etmektedir. MPC ne kadar yüksekse orijinal veriye o kadar yakın olmakta. MaksSA, MAD ne kadar düşük ise geri elde edilen veri orijinale o kadar yakın olmaktadır.

### **4.3. Modifiye Edilmiş Sözlük Ve Seyrek Gösterimle Süperpiksel Temelli Anomali Düzeltmeli Sıkıştırma Yöntemi (MSSASR-RX)**

Verideki anomali kısımlar verinin geneline uymayan bölgelerdir. MSSASR-RX yöntemi süperpiksel tabanlı olup seyrek katsayı hesabında MSSASR algoritmasını kullanmaktadır. MSSASR algoritmasının verimli çalışma bilmesi için süperpiksel içerisinde bulunan imzalar birbirine benzer olmalıdır. Fakat anomali pikseller içerisinde bulunduğu süperpikselle aynı karakteristiği göstermemektedir. Bundan

dolayısıyla seyrek katsayı hesabında ilgili anomali pikseller diğer piksellere göre daha kötü sıkıştırılmaktadır.



Şekil 4.22. MSSASR-RX yönteminin işlem aşamaları

Bu sorunu engellemek ve anomali noktaların daha iyi korunmasını sağlamak amacıyla MSSASR-RX yöntemi önerilmektedir. MSSASR-RX yöntemi verideki anomali pikselleri korumak için SSASR yönteminin özelleştirilmiş halidir. Bu sebeple anomali yerleri SSASR'a göre daha iyi korumaktadır. MSSASR-RX yönteminin bir diğer avantajlı tarafı ise anomalileri çok iyi korumakla birlikte diğer bütün metrik sonuçlarında da SSASR yöntemine göre önde olmaktadır. Önerilen yöntemin işlem basamakları Şekil 4.22'te gösterilmektedir.

MSSASR-RX yöntemi, anomali olan ve olmayan piksellere ayrı ayrı sıkıştırma uygulamaktadır. Bu sebeple yöntemin başlangıcında anomali olan pikseller tespit edilmektedir. Anomali olan pikseller için sıkıştırma aşamaları şu şekildedir.

- Anomali piksellerin bulunması.
- Vektörel dönüşüm yardımıyla spektral matris haline çevrilmesi.
- PCA yardımıyla sözlüğün oluşturulması.
- SOMP algoritması yardımıyla seyrek katsayı hesabının yapılması.
- Sözlüğün ve anomali haritasının entropi kodlanması.

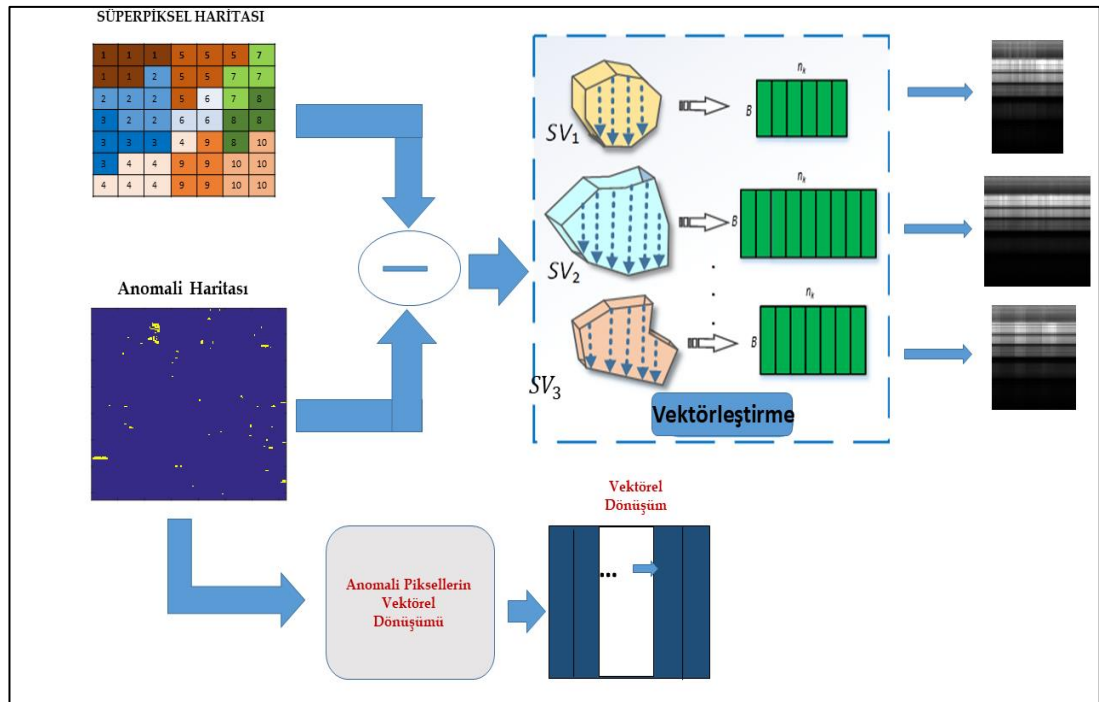
- Seyrek katsayılar nicemlendikten sonra DPCM ve entropi sıkıştırırmaya dahil olması.

Anomali olmayan piksellerin sıkıştırma aşamaları ise şu şekildedir.

- Anomali olmayan piksellerin tespiti.
- Süper piksel bölütleme işlemi.
- Vektörel dönüşüm işlemi.
- Sözlük öğrenmesi.
- Seyrek katsayıların bulunması.
- Seyrek katsayı matrisinin sıralı olarak birleştirilmesi.
- Seyrek katsayılar nicemlendikten sonra DPCM ve entropi sıkıştırırmaya dahil olması.
- Sözlüğün entropi kodlanması.

#### 4.3.1. Süperpiksel haritasının bulunması ve vektörleştirme işlemi

Süperpiksel haritası Bölüm 3'te algoritması verilen SLIC yöntemiyle hesaplanmaktadır.



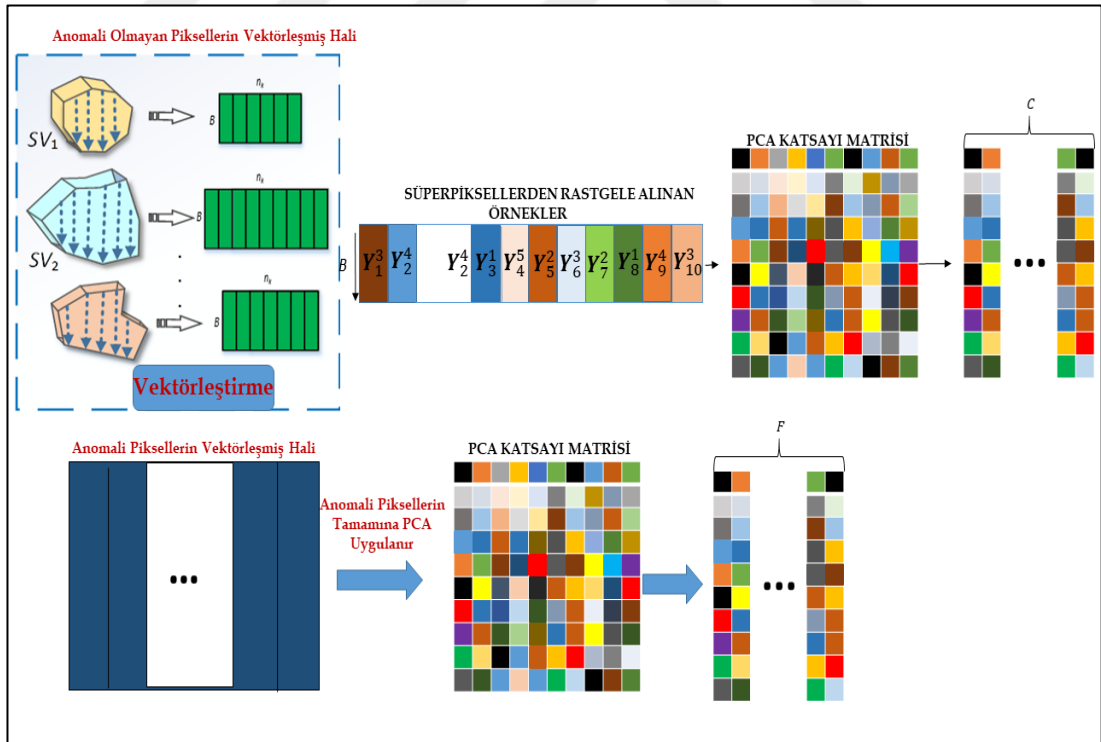
Şekil 4.23. Anomali olan ve olmayan pikseller için vektörel dönüşüm

SLIC yöntemi hızlı olduğu için ve süperpiksellerin kompakt olmasını sağladığı için tercih edilmektedir. Süperpikseller oluşturulduktan sonra vektörel dönüşüm işlemi yapılırken, anomali olan pikseller kendi içinde ayrıca vektörel dönüşüm işlemine tabi tutulmaktadırlar. Anomali olmayan piksellerin vektörel dönüşümü şu şekilde yapılmaktadır. İlgili süperpiksel içerisindeki, anomali pikseller çıkarıldıktan sonra yukardan aşağıya doğru zikzak taramayla vektörel dönüşüm yapılmaktadır.

Şekil 4.23'te gösterilen çıkarma işlemi, ilgili süperpikselden ona karşılık gelen anomali pikselin çıkarılması manasına gelmektedir.

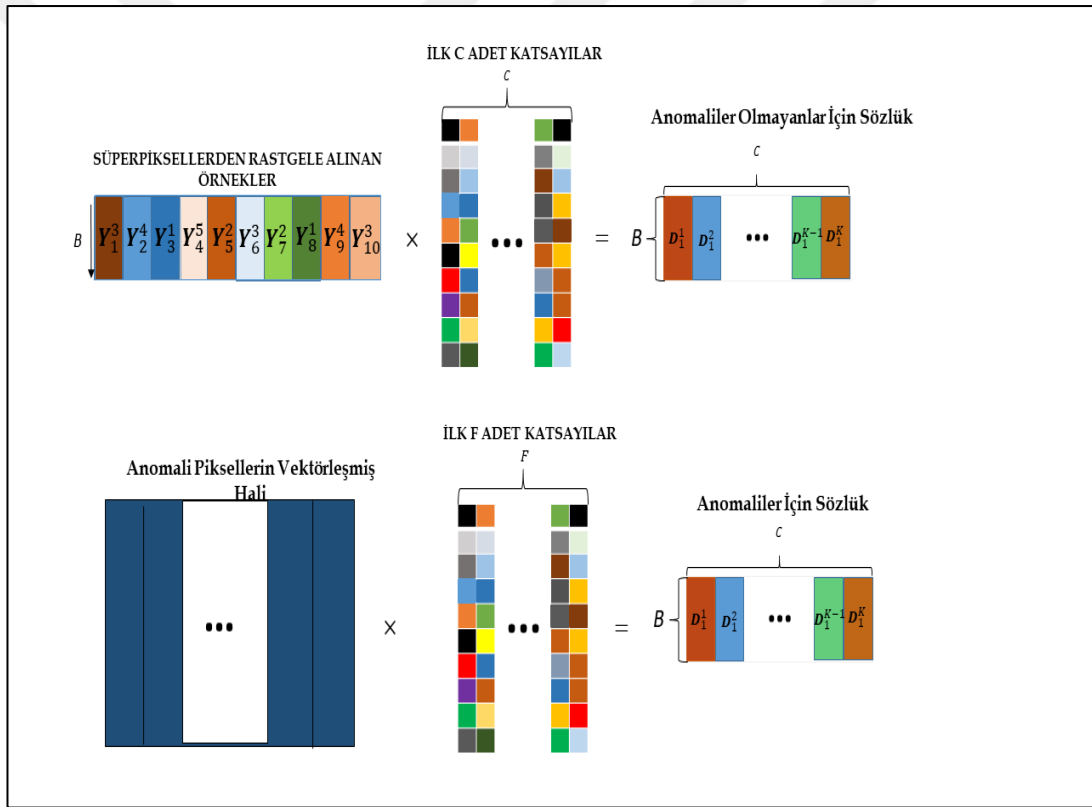
#### 4.3.2. Sözlük öğrenmesi işlemi

Sözlük öğrenmesi işlemi iki aşamada gerçekleşmektedir. Birinci aşama, anomali pikseller için sözlük oluşturma işlemidir. Anomali pikseller için sözlük öğrenme işleminde, ilk olarak anomali olan tüm pikseller üzerinden PCA dönüşümü alınarak PCA katsayı matrisi oluşturulmaktadır.



Şekil 4.24. MSSASR-RX için sözlük öğrenme işlem basamakları

Ardından Şekil 4.24’da gösterildiği gibi ilk F adet katsayı sözlük atomlarını oluşturmak için seçilmektedir. Buradaki F değeri anomali pikselleri ne kadar iyi sıkıştırmak istediğimize bağlı olarak değişmektedir. Son aşamada, Şekil 4.25’de gösterildiği gibi F adet PCA katsayısı ile vektörel dönüşüm sonucu elde edilen anomali pikseller çarpılarak sözlük için atomlar öz uzayda oluşturulur. İkinci aşama sözlük öğrenme işlemi işleminde PCA yöntemi kullanılmaktadır. İlk olarak, K adet süperpikselin her birinden T adet piksel seçilmektedir. Her bir süperpikselden seçilen T adet örnek Şekil 4.25’de verildiği gibi yan yana dizilerek birleştirilmektedir. Birleştirme işlemi sonrası oluşturulan matrise, PCA uygulanarak temel bileşen katsayı matrisi oluşturulmaktadır.



Şekil 4.25. MSSASR-RX için sözlük öğrenme işlem basamakları

PCA işlemi sonucu hesaplanan öz vektörlerden en değerli C adet öz vektör seçilmektedir. Burada C, sözlük boyutunu belirleyen kontrol parametresidir. SSASR metodunda C parametresi belirlenirken C'nin boyutu spektral bant genişliğinin 3/2 katı alınmaktadır. Bundan dolayı SSASR’da sözlük oluşturulurken, her bir alt sözlükten eşit sayıda atom seçilerek spektral bant genişliğinin 3/2 katı toplamda atom

seçilir. MSSASR-RX’de ise C parametresi, Şekil 4.5(b)’de görüldüğü gibi sözlük atomu kullanım histogramı ilk atomlarda yoğunlaştığı için bant genişliğinin yarısı kadar seçilmek yeterli olmaktadır. Örneklenmiş veri matrisi, C’ye bağlı olarak seçilen öz vektörler ile Şekil 4.25’de gösterildiği gibi PCA uzayına taşınır ve bu sayede öz uzayda sözlük için atomlar oluşturulmuş olur. Şekil 4.24 ve Şekil 4.25’de MSSASR-RX için sözlük öğrenme aşamaları gösterilmektedir.

Öz uzayda oluşturulan sözlük atomları, seyrek katsayı oluşturma aşamasında kullanılmaktadır. Sözlük atomlarının kullanım sayısı, sıkıştırma oranına göre değişim göstermektedir. 0,1, 0,2 bpp (bit per pixel) gibi düşük sıkıştırma oranlarında az sayıda sözlük atomuna ihtiyaç duyulurken 0,9, 1 bpp gibi yüksek sıkıştırma oranlarına çıktıkça ihtiyaç duyulan sözlük atomu sayısı artmaktadır. Bundan dolayı, her sıkıştırma oranı için sözlüğün hepsini alıcıya göndermeye ihtiyaç yoktur. Az kullanılan sözlük atomlarını atmak için sözlük atomlarının kullanım sayısı tespit edilerek, belirli bir kullanım oranının altında olan sözlük atomları atılmaktadır.

### **4.3.3. Seyrek katsayıların belirlenmesi**

Seyrek tanımlama, sıkıştırılacak veriyi kullanılan sözlüğe bağlı olarak katsayılarının çoğu sıfırlardan oluşan matris şeklinde ifade etme işlemidir. Bu matris içerisinde bulunan katsayıların çoğunluğu, sıfır olduğu için bu verinin sıkıştırma işlemi verimli olmaktadır. Süperpikseller ilintili pikselleri gruplandığı için eşzamanlı seyrek katsayı modeliyle tanımlanabilmektedir. MSSASR-RX yöntemi süperpiksel temelli olduğu için SOMP algoritmasını kullanmaktadır. Seyrek katsayı tanımlama işlemi Bölüm 4.1’deki yöntemle aynı olduğu için burada tekrardan anlatılmamıştır.

### **4.3.4. Nicemleme entropi kodlama ve kod çözme**

Sıkıştırıp gönderirken takip edilen aşamalar şöyle özetlenebilir. İlk olarak anomali haritası, anomali olan ve olmayan pikseller için oluşturulan sözlükler entropi kodlanmaktadır. Entropi kodlama işleminde Huffman kodlama kullanılmaktadır [25]. Seyrek katsayıları daha iyi sıkıştırılabilmek için Huffman kodlama öncesinde nicemleme ve DPCM işlemine tabi tutulmaktadır. Nicemleme işleminde tekdüze nicemleyici kullanılmaktadır. Veriyi geri elde ederken Huffman kod çözücü

kullanılmaktadır. Elde edilen seyrek katsayılarla sözlük atomları çarpılarak geri elde edilen değerler anomali haritasına göre sıralanıp geri elde etme işlemi tamamlanmaktadır.

#### **4.3.5. Sıkıştırma performansının güncel yöntemlerle karşılaştırılması**

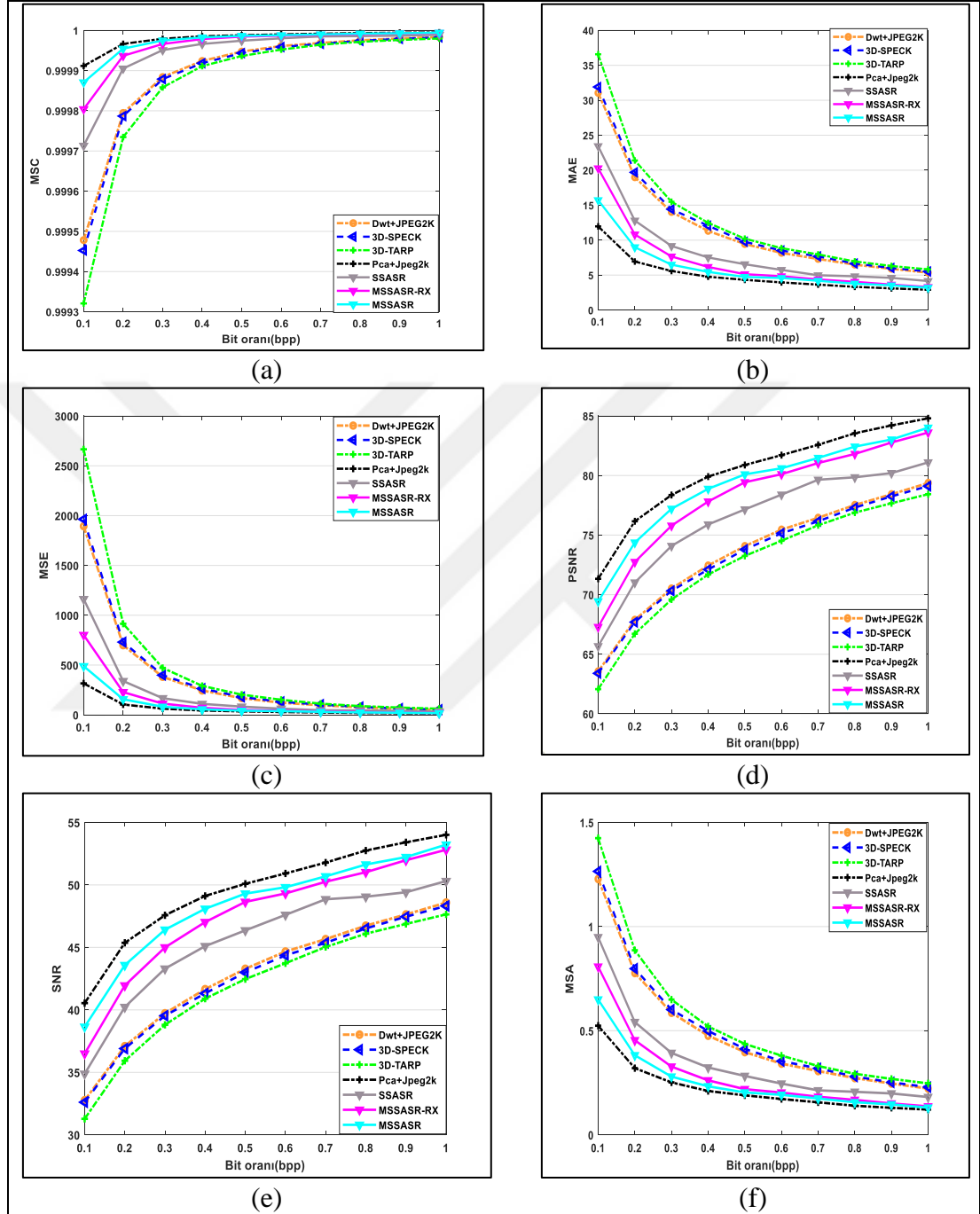
Önerilen MSSASR-RX yöntemi ve DWT-JP2K, 3D-SPECK, 3D-TARP, SSASR yöntemlerinin farklı metriklerdeki bit oranına bağlı grafikleri Moffett Field için Şekil 4.26(a), Şekil 4.26(b), Şekil 4.26(c), Şekil 4.26(d), Şekil 4.26(e) ve Şekil 4.26(f)'de verilmektedir. Indian Pines için Şekil 4.27(a), Şekil 4.27(b), Şekil 4.27(c), Şekil 4.27(d), Şekil 4.27(e) ve Şekil 4.27(f)'de verilmektedir. Jasper Ridge için Şekil 4.28(a), Şekil 4.28(b), Şekil 4.28(c), Şekil 4.28(d), Şekil 4.28(e) ve Şekil 4.28(f)'de verilmektedir. Washington DC için Şekil 4.29(a), Şekil 4.29(b), Şekil 4.29(c), Şekil 4.29(d), Şekil 4.29(e) ve Şekil 4.29(f)'de verilmektedir.

MSSASR-RX yöntemi için süperpiksel sayısı belirlenirken MSSASR yöntemindeki mantık kullanılmıştır. Buna göre veri boyutu büyüdükçe gerekli olan süperpiksel sayısı artmaktadır. Süperpiksel başına düşen ortalama piksel sayısı 43 pps (pixel per superpixel) olduğu zaman, veriyi ifade etmek için yeterli olduğu görülmektedir. Sözlük oluşturmada fazla işlem yükünden kaçmak için her bir süperpikselden 3 örnek almak yeterli olmaktadır. Önerilen yöntemin anomali pikselleri koruyup korumadığını bulmak için Moffett Field, Jasper Ridge, Indian Pines ve Washington DC verilerine RX anomali tespiti yöntemi uygulanarak anomali haritası elde edilmektedir. Bu haritadan, anomali değeri en yüksek 150 piksel bütün veriler için seçilmektedir. Veriler farklı yöntemlere göre sıkıştırılıp iletildikten sonra her bir yöntem için daha önce tespit edilen anomali pikseller üzerinden ortalama SNR değerlerine bakıldığı zaman Şekil 4.30'de gösterildiği gibi MSSASR-RX yöntemi diğer yöntemlere kıyasen başarılı olmaktadır.

Sıkıştırma sonrasında geri elde edilen verinin genel kalitesini bozunumunu en iyi özetleyen metrikler SNR, MSA ve MSC metrikleridir. Grafiklerin tamamında MSSASR-RX yöntemi PCA-JPEG2K hariç diğer tüm yöntemlerden daha iyi bir performans sergilediği görülmektedir. SNR, PSNR ve MSC metriklerinde orijinal



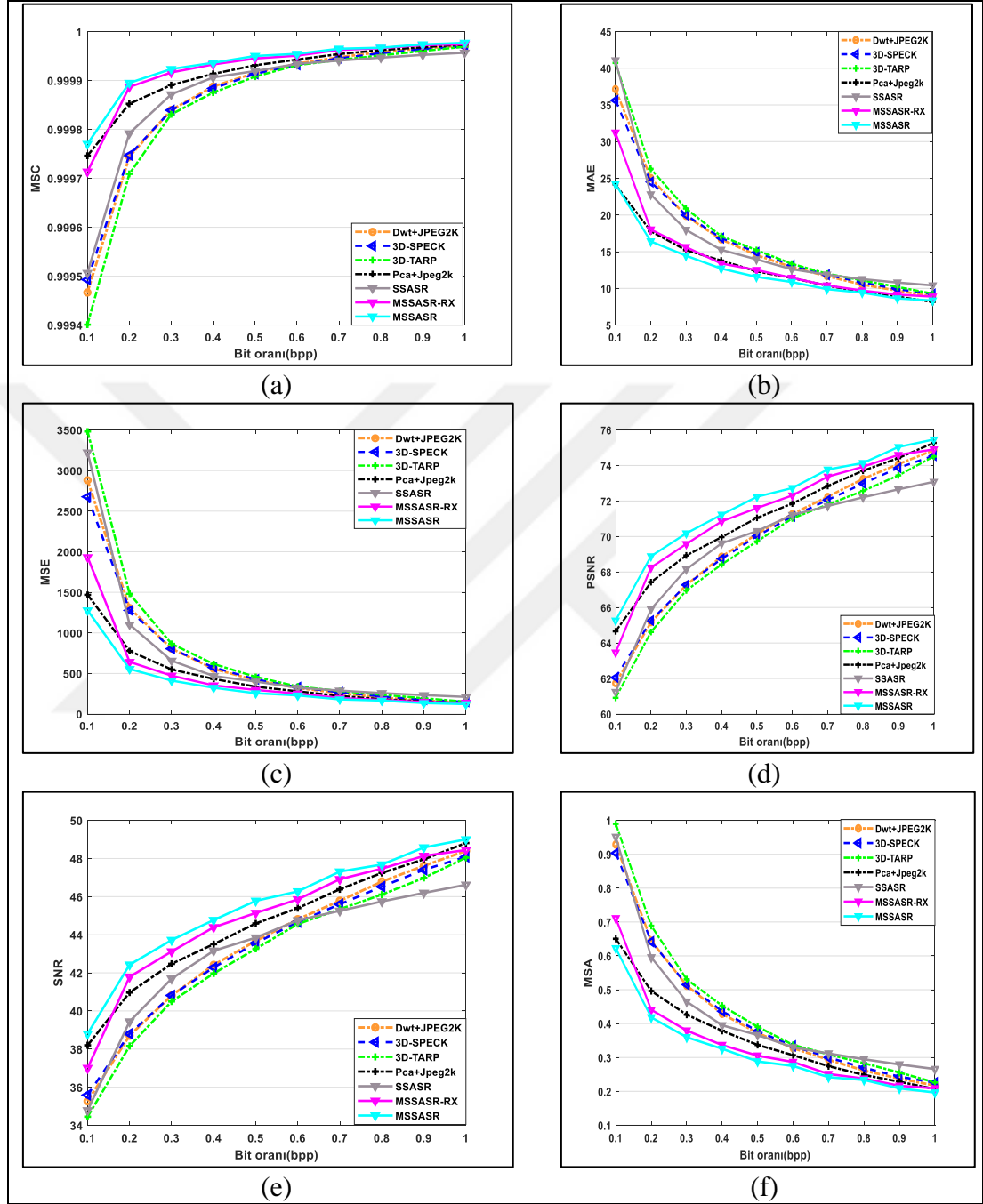
veriye yakınlaşmak için istenen sonuçların yüksek olmasıdır. MSA, MAE, MAE metriklerinde ise ne kadar düşük ise orijinal veriye o kadar yakın olmaktadır.



Şekil 4.26. Moffett Field için metrik sonuçları

Şekil 4.26’da MSSASR-RX yönteminin Moffett Field için metrik sonuçları verilmiştir. PCA+JPEG2K ve MSSASR yöntemi hariç diğer yöntemler karşısında tüm parametreler için önde olmaktadır. PCA+JPEG2K yöntemi karşısında ise yüksek

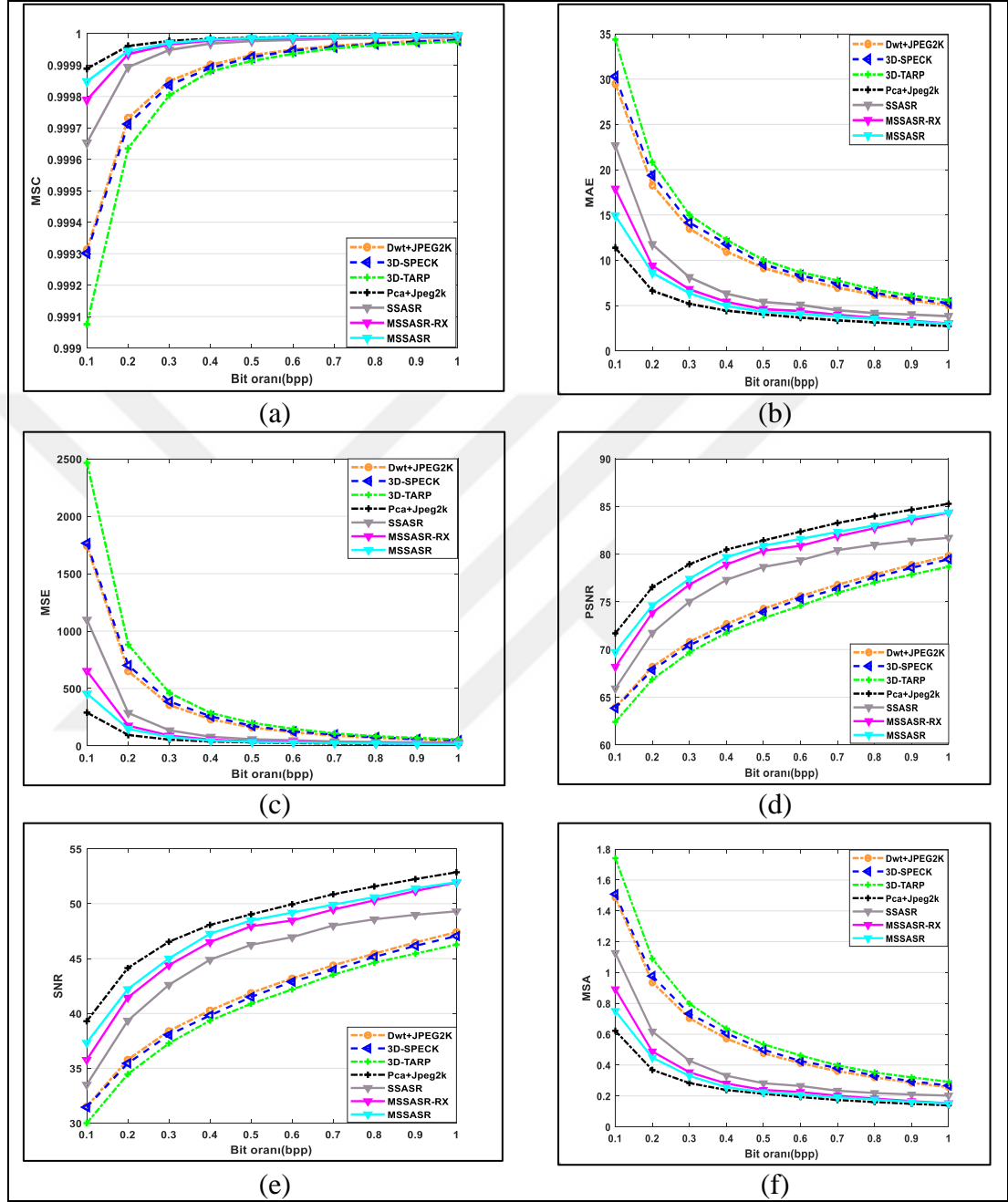
bit oranlarında yaklaşık aynı seviyede olmaktadır. MSSASR ile yaklaşık aynı seviyede olmaktadır.



Şekil 4.27. Indian Pines için metrik sonuçları

Şekil 4.27’de MSSASR-RX yönteminin Indian Pines için metrik sonuçları verilmiştir. MSSASR yöntemi hariç bütün yöntemler karşısında tüm parametreler

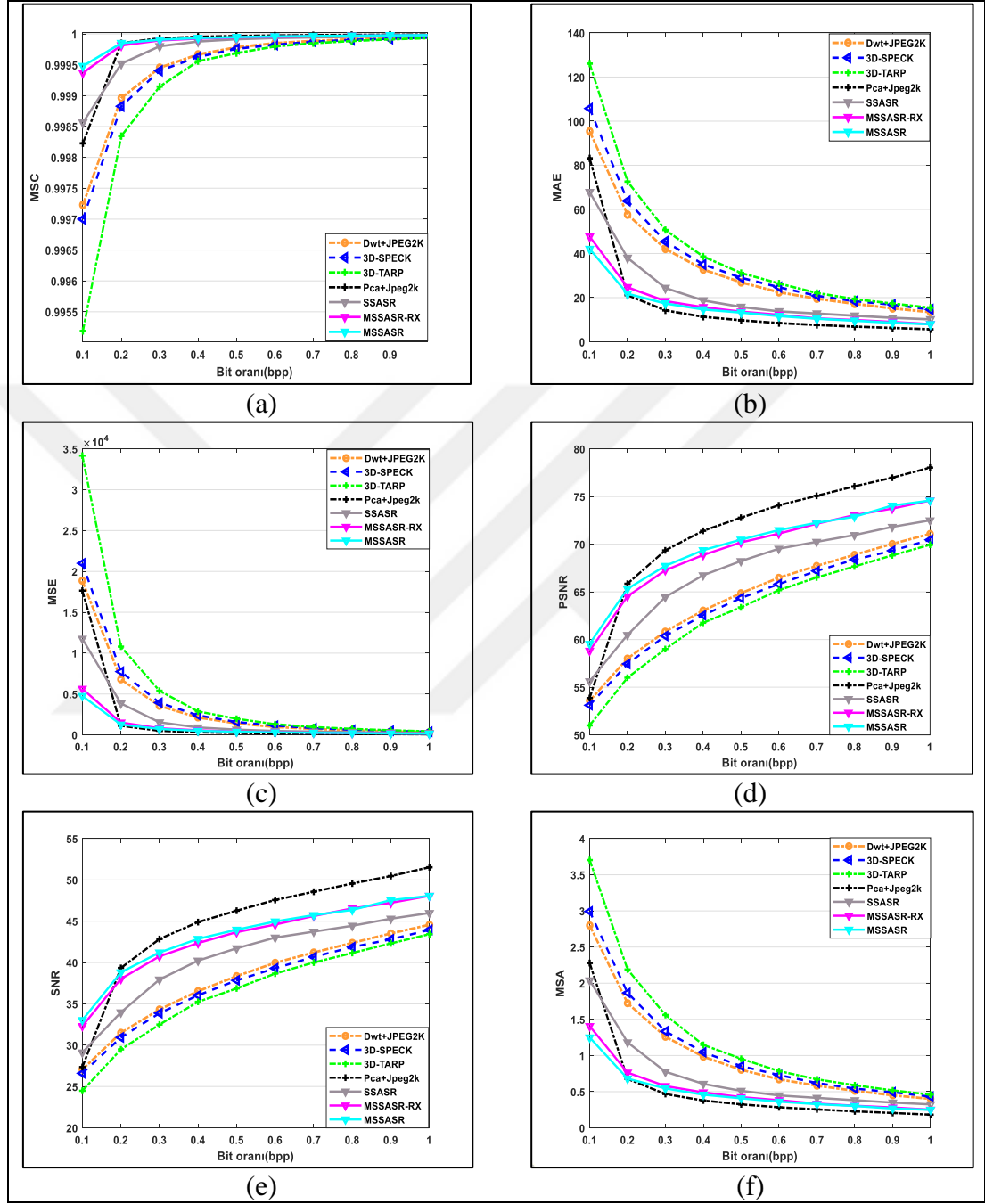
için önde olmaktadır. PCA+JPEG2K yöntemi karşısında 0,1 bit seviyelerinde geride olmaktadır.



Şekil 4.28. Jasper Ridge için metrik sonuçları

Şekil 4.28’de MSSASR-RX yönteminin Jasper Ridge için metrik sonuçları verilmiştir. PCA+JPEG2K ve MSSASR yöntemi hariç diğer yöntemler karşısında tüm parametreler için önde olmaktadır. PCA+JPEG2K yöntemi karşısında ise yüksek

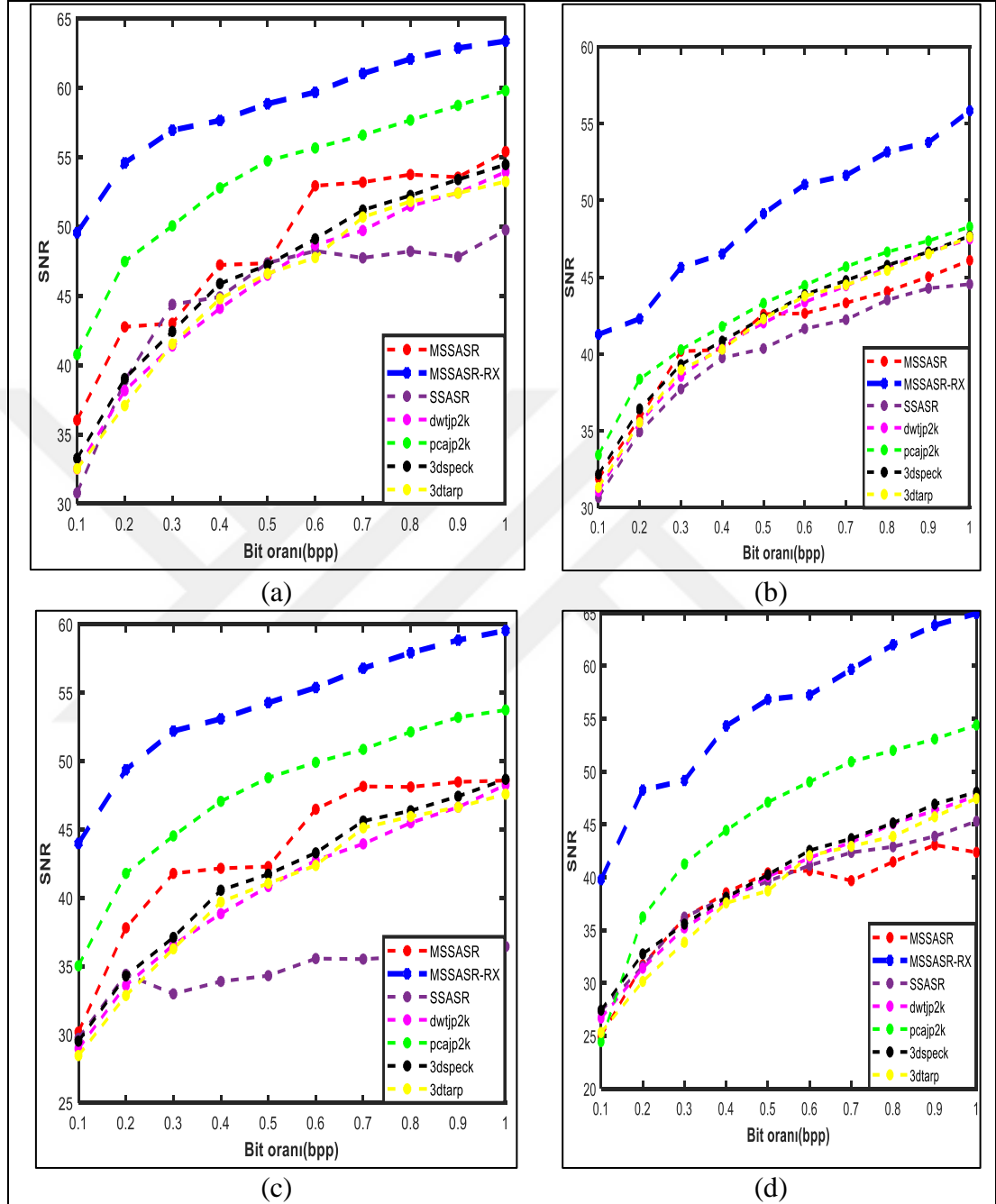
bit oranlarında yaklaşık aynı seviyede olmaktadır. MSSASR ile yaklaşık aynı seviyede olmaktadır.



Şekil 4.29. Washington DC için metrik sonuçları

Şekil 4.29’da MSSASR-RX yönteminin Washington DC için metrik sonuçları verilmiştir. PCA+JPEG2K ve MSSASR yöntemi hariç diğer yöntemler karşısında

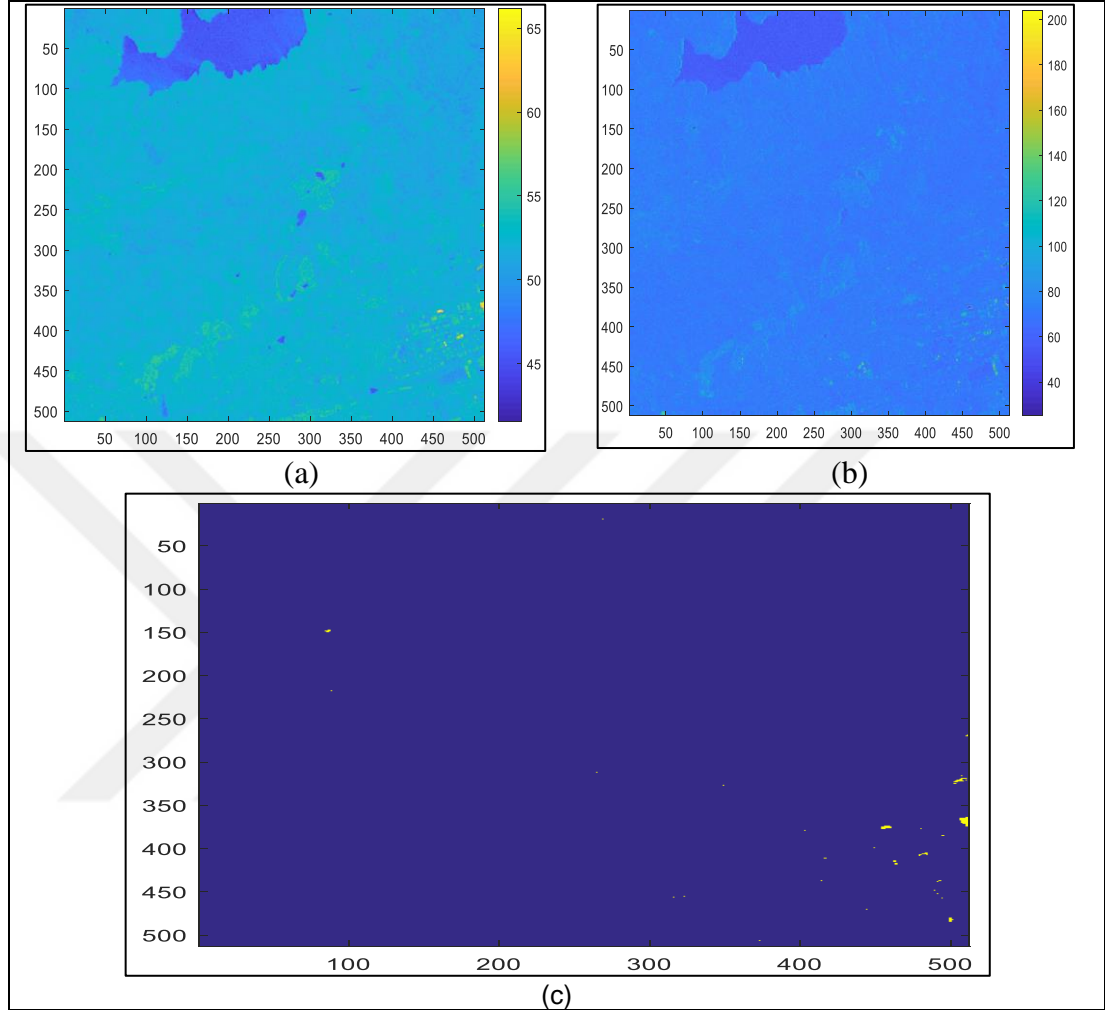
tüm parametreler için önde olmaktadır. PCA+JPEG2K yöntemi karşısında ise düşük bit oranlarında önde olmaktadır. MSSASR ile yaklaşık aynı seviyede olmaktadır.



Şekil 4.30. 150 anomali piksel için dört farklı veride ortalama SNR değeri

Şekil 4.30'da, 150 anomali piksel için dört farklı veride ortalama SNR değeri verilmektedir. Moffett Field için Şekil 4.30(a), Indian Pines için Şekil 4.30(b), Jasper Ridge için Şekil 4.30(c) son olarak Washington DC için Şekil 4.30(d)'de SNR

dağılımları gösterilmektedir. Önerilen yöntem verilen bütün yöntemler karşısında SNR açısından önde olmaktadır.



Şekil 4.31. Moffett verisi için (a)'da SNR dağılım haritası (b)'de hata haritası verilmiştir (c)'de ise anomali dağılım haritası verilmektedir

Şekil 4.31(a)'da Moffett verisi 0,9 bit oranında SNR dağılım haritası ve Şekil 4.31(b)'de hata haritası verilmektedir. Anomali haritası 150 piksel için Şekil 4.31(c)'de gösterilmiştir. MSSASR-RX yöntemi ile sıkıştırma sonucunda Şekil 4.31(a)'da verildiği gibi anomali piksellerin SNR'ı yüksek olduğu için koyu sarı bölgeler ile gösterilmektedir. MSSASR-RX yöntemi ile sıkıştırma sonucunda Şekil 4.31(b)'de verildiği gibi anomali pikseller daha iyi korunduğu için hatalar düşük olmaktadır. Bu sebeple hata haritasında koyu mavi bölgeler ile gösterilmektedir.

## 5. SONUÇLAR VE ÖNERİLER

MSSASR yöntemi, hiperspektral veriler için SSASR temelli yeni bir kayıplı sıkıştırma yöntemi sunmaktadır. Bu yöntem, sözlük oluşturmak için çoklu PCA dönüşümü yerine tekli PCA dönüşümü kullanmaktadır. Sözlükte yapılan bu güncelleme sayesinde seyrek katsayıların enerjisinin %98'si ilk yedi satırda toplanmaktadır. Bu sayede, seyrek katsayılar daha iyi bir şekilde ifade edilmektedir. Ayrıca, önerilen yöntem süperpiksel haritası kodlama ihtiyacını ortadan kaldırdığı için fazladan bit yükünü ortadan kaldırmaktadır. MSSASR yöntemi, SSASR metodunu MAE, MSE, MSA, SNR, MSC, PSNR gibi farklı karşılaştırma metriklerde daha iyi değerler elde edilmesini sağlamıştır. Bu sayede, spektral bozulmayı daha aza indirmektedir. Deneysel çalışmalar kısmında, farklı sıkıştırma oranları için farklı metrik sonuçları ayrıntılı bir şekilde verilmiştir. MSSASR yöntemi, genel olarak literatürdeki güncel yöntemlerden daha iyi sıkıştırma performansı sergilemektedir.

ECSSASR çalışmasında, hiperspektral görüntülerin seyrek katsayılar yardımıyla sıkıştırılması için yeni bir yöntem önerilmiştir. Önerilen yöntemde veri ilk olarak bütün bir şekilde sözlük ve katsayılar yardımıyla sıkıştırılmaktadır. İlk sıkıştırılmadan sonra geri elde edilen verinin SNR dağılım haritasına bakılarak, elde mevcut olan sözlüğün hangi pikselleri ifade edemediği tespit edilmektedir. Yüksek hata ile geri elde edilen pikseller bir sonraki aşamada ayrı bir veriymiş gibi düşünülerek kendine ait sözlüğü ve katsayıları hesaplanarak ayrıca gönderilmektedir. Önerilen yöntem sayesinde hatalı olan pikseller düzeltilmiş olmaktadır.

MSSASR-RX yöntemi yenilenmiş sözlüğün avantajını kullanmaktadır. Ayrıca anomali pikseller için oluşturduğu yeni sözlük yardımıyla anomali bölgeleri çok iyi korumaktadır. MSSASR-RX yöntemi MSSASR ve ECSSASR yönteminin avantajlarını birleştirmektedir. Çünkü anomali pikseller süperpiksel homojenliğini bozduğu için ifade edilme oranı düşüktür. Anomali pikseller için, kendilerine özgü sözlük oluşturulduğu zaman otomatik olarak ifade edilme oranı artmaktadır. Böylece hem SNR'da iyileşme hem de anomali koruma birlikte sağlanmaktadır.

Sistemler hız yönünden incelendiğinde ECSSASR yöntemi SSASR yöntemine göre daha yavaştır çünkü fazladan işlem yükü olarak hatalı pikselleri tespit etmekte ve onlara uygun yeni bir sözlük oluşturmaktadır. MSSASR yöntemi SSASR yönteminden hız bakımından daha iyidir çünkü sözlük oluşumunda sadece bir defa PCA almaktadır. MSSASR-RX yöntemi MSSASR yönteminden daha yavaştır çünkü anomali fazladan anomali piksellerin tespitini yapmakta ve onlara uygun sözlük oluşturmaktadır. Sözlük öğrenme temelli olan yöntemlerin başarımı sözlük ve seyrek katsayılara bağlı olduğu için gelecek çalışmalarda, sözlük ve katsayı seçiminde iyileştirilme yapılması planlanmaktadır.





## KAYNAKLAR

- [1] [http://www.ktu.edu.tr/dosyalar/ormanamenajmani\\_3fc8b.pdf](http://www.ktu.edu.tr/dosyalar/ormanamenajmani_3fc8b.pdf) (Ziyaret Tarihi: 16 Aralık 2019)
- [2] <https://www.bilimcag.com/nedir/hiperspektral-goruntuleme-nedir/> (Ziyaret Tarihi: 16 Aralık 2019).
- [3] <https://usimsek.files.wordpress.com/2011/12/caps4.png10> (Ziyaret Tarihi: 27 Aralık 2019).
- [4] <https://haritaonline.blogspot.com/2014/06/uzaktan-alglamada-sensor-sistemleri.html> (Ziyaret Tarihi: 16 Aralık 2019).
- [5] Fu W., Li S., Fang Li., Benediktsson J.A., Adaptive Spectral-Spatial Compression of Hyperspectral Image with Sparse Representation, *IEEE Trans. Geosci. Remote Sens.*, 2017, **55**(2), 671–682.
- [6] Chuan F., Yaohua Y., Fulin L., Bo D., Hyperspectral Image Compression Based on Simultaneous Sparse Representation and General-pixels, *Pattern Recognition Letters*, Eylül 2018.
- [7] Karaca A.C., Ertürk A., Güllü M.K., Elmas M., Ertürk S., Hiperspektral Görüntüleme İle Adli Belgelerdeki Bulguların Analizi, *20. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU2012)*, Antalya, Türkiye, 18-20 Nisan
- [8] Karaca A.C., Hiperspektral Görüntülerin Tahmin Temelli Sıkıştırılması İçin Tahmin Temelli Yaklaşımlar, Doktora Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2019, 575130.
- [9] Du Q., Fowler J.E., Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis, *IEEE Geosci. and Remote Sens. Lett.*, 2007, **4**(2), 201–205.
- [10] Artuğer F., Özkaynak F., JPEG Sıkıştırma Algoritmasının Dünü Bugünü ve Geleceği, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 30(3), 161-167.
- [11] Rucker J.T., Fowler J.E., Younan N.H., JPEG2000 Coding Strategies for Hyperspectral Data In Proceedings, *IEEE International Geoscience and Remote Sensing Symposium*, 2005, **1**(4).

- [12] Alaydın J.G., Töreyn B.U., Sparse Coding Based Compression of Spectrally Uncorrelated Hyperspectral Data Using Haar Wavelet Transform, *24th Signal Processing and Communication Application Conference (SIU)*, 2016, 1945-1948.
- [13] Karaca A.C, Ertürk A., Güllü M.K., Ertürk S., Kamufraj Tespiti için Hiperspektral Görüntüleme/Hyperspectral Imaging for Camouflage Detection. *EMO BİLİMSEL DERGİ*, 2013, **3**(5), 35-39.
- [14] Şimşek M., Polat E., Sparse Representation-Based Dictionary Learning Methods for Hyperspectral Super-Resolution, *IEEE 24th Signal Processing and Communication Application Conference (SIU)*, 2016, 753-756.
- [15] Mesut A., Veri Sıkıştımda Yeni Yöntemler, Doktora Tezi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Edirne, 2006
- [16] Bayır Ö., Analiz ve Sentez Seyreklik İçin Sözlük Öğrenme Algoritmaları ve Görüntü İşlemede Uygulamaları, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2015.
- [17] Aharon M., Elad M., Bruckstein A., K-SVD: An Algorithm For Designing Overcomplete Dictionaries for Sparse Representation, *Signal Processing, IEEE Transactions*, Kasım 2006, **54**(11), 4311-4322.
- [18] Engan K., Aase S.O., Husoy J.H., Method of Optimal Directions For Frame Design, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, 2443-2446.
- [19] Xu H., Wang X.J., Applications of Multispectral/Hyperspectral Imaging Technologies In Military, *Infrared and Laser Engineering*, 2007, **36**(1),13-18.
- [20] Achanta R., Shaji A., Smith K., Lucci A., Fua P., Süsstrunk S., SLIC Superpixels Compared to State-of-the-Art Superpixel Methods, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 2012, **34**(11), 2274–2282
- [21] Dong W., Zhang L., Shi G., and Li X., Nonlocally Centralized Sparse Representation for Image Restoration, *IEEE Trans. Image Process.*, 2013, **22**(4), 1620–1630.
- [22] Li J., Zhang H., and Zhang L., Efficient Superpixel-Level Multitask Joint Sparse Representation For Hyperspectral Image Classification, *IEEE Trans. Geosci. Remote Sens.*, 2015, **53**(10), 5338–5351.
- [23] Huang J., Zhang T., Metaxas D., Learning with Structured Sparsity, in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, 417–424.
- [24] Huffman D.A., A Method for Construction of Minimum-Redundancy Codes, *Proc. IRE*, 1952 **40**(9), 1098–1101.

- [25] Tropp J.A., Gilbert A.C. Strauss M.J., Simultaneous Sparse Approximation Via Greedy Pursuit, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, USA, 2005.
- [26] Du Q. Fowler J.E., Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis, *IEEE Geosci. and Remote Sens. Lett.*, 2007, **4**(2):201–205.
- [27] Xiaoli T., Pearlman W.A., Three-Dimensional Wavelet-Based Compression of Hyperspectral Images, *Hyperspectral Data Compression Springer Boston*, 273–308.
- [28] Wang Y., Rucker J.T., Fowler J.E., 3D Tarp Coding for the Compression of Hyperspectral Images, *IEEE Geosci. and Remote Sens. Lett.*, 2014, **1**(2):136–140.
- [29] Küçük F., Töreyn B.U., Çelebi F.V., Sparse And Low-Rank Matrix Decomposition-Based Method for Hyperspectral Anomaly Detection, *Journal of Applied Remote Sensing*, 2019, **13**(1), 014-519.
- [30] Gündoğar Z., Töreyn B.U., Demiralp M., Tridiagonal Folmat Enhanced Multivariance Products Representation Based Hyperspectral Data Compression, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018, **11**(9), 3272-3278.
- [31] Alaydın J.G., Töreyn B.U., Sparse Coding Based Compression of Spectrally Uncorrelated Hyperspectral Data Using Haar Wavelet Transform, *24th Signal Processing and Communication Application Conference (SIU)*, 2016, 1945-1948.
- [32] Ülkü İ., Töreyn B.U., Sparse Representations for Online-Learning-Based Hyperspectral Image Compression, *Applied optics*, 2015, **54**(29), 8625-8631.
- [33] Ülkü İ., Töreyn B.U., Sparse Coding of Hyperspectral Imagery Using Online Learning, *Signal Image and Video Processing*, 2015, **9**(4), 959-966.

## KİŞİSEL YAYIN VE ESERLER

- [1] **Ertem A.**, Karaca A.C., Güllü M.K., Compression of Hyperspectral Images Using Superpixel Based and Error-Corrected Sparse Representation. *27th Signal Processing and Communications Applications Conference*, Sivas, Türkiye, 24-26 Nisan 2019.



## ÖZGEÇMİŞ

Şubat 1992’de Tortum/Erzurum’da doğdu. 2006’da ilk ve ortaokulu Erzurum - Tortum Uncular İlköğretim Okulunda ve lise öğrenimini ise Erzurum-Merkez Anadolu Lisesi’nde 2010 yılında tamamladı. 2011 yılında Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği bölümünü kazandı ve 2016 yılında Elektronik ve Haberleşme Mühendisi olarak mezun oldu. Şubat 2017’de Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Anabilim Dalı’nda Yüksek Lisans eğitimine başladı.

