

**T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY SİNİR AĞLARI VE RASTGELE ORMAN YÖNTEMLERİ İLE LANDSAT 8  
GÖRÜNTÜLERİNDEN OTOMATİK KIYI ÇİZGİSİ ÇIKARTILMASI**

**ABDULKADİR İNCE**

**YÜKSEK LİSANS TEZİ  
HARİTA MÜHENDİSLİĞİ ANABİLİM DALI  
UZAKTAN ALGILAMA VE CBS PROGRAMI**

**DANIŞMAN  
PROF. DR. BÜLENT BAYRAM**

**İSTANBUL, 2018**

T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY SİNİR AĞLARI VE RASTGELE ORMAN YÖNTEMLERİ İLE LANDSAT 8  
GÖRÜNTÜLERİNDEN OTOMATİK KIYI ÇİZGİSİ ÇIKARTILMASI**

Abdulkadir İNCE tarafından hazırlanan tez çalışması 22/03/2018 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Prof. Dr. Bülent BAYRAM  
Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Prof. Dr. Bülent BAYRAM  
Yıldız Teknik Üniversitesi

Prof. Dr. Dursun Zafer ŞEKER  
İstanbul Teknik Üniversitesi

Dr. Öğr. Ü. Alper ŞEN  
Yıldız Teknik Üniversitesi

---

---

---



Sunulan tez çalışması, Prof. Dr. Bülent Bayram' ın yürütücülüğünü yaptığı 115Y718 numaralı ve "SÜRDÜRÜLEBİLİR KIYI ALANI İZLEME MODELİ İÇİN İHA VE İHA-LİDAR VERİLERİNDEN OTOMATİK ÜÇ BOYUTLU KIYI ÇİZGİSİ ÇIKARTILMASI VE ANALİZİ: TERKOS (İSTANBUL) ÖRNEĞİ" başlıklı TÜBİTAK 1001 projesi ile desteklenmiştir.

## ÖNSÖZ

---

Sunulan tez çalışmasında denetimsiz öğrenme yöntemi olan kendini düzenleyen haritalar ile denetimli öğrenme yöntemleri olan yapay sinir ağları ve rastgele orman yöntemleri kıyı çizgisi çıkarımında kullanılarak sonuçlar karşılaştırılmıştır. Daha sonra kendini düzenleyen haritalar yöntemi sonucunda oluşan deniz-kara sınıfları yapay sinir ağları ve rastgele orman yöntemlerinde girdi olarak kullanılmış ve bu şekilde de kıyı çizgisi çıkartılmıştır. Görüntü işleme tekniklerinin kullanıldığı bu çalışmada, denetimsiz öğrenme yöntemi sonuçları denetimli öğrenme yöntemlerinde kullanılarak kullanıcı kaynaklı hataları elimine etmek, denetimli öğrenme yöntemlerinde eğitim için veri toplama aşamasını ortadan kaldırarak zamandan tasarruf sağlamak ve elde edilen sonuçlarda doğruluk değerlerinin artırılması amaçlanmıştır.

Tez çalışmamın planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren, sayın hocam Prof. Dr. Bülent BAYRAM' a sonsuz teşekkürlerimi sunarım.

Çalışma süresince bilgi, görüş ve önerilerinden yararlandığım sayın hocalarım Prof. Dr. Dursun Zafer ŞEKER ve Öğr. Ü. Alper ŞEN' e teşekkür ederim.

Çalışma arkadaşlarım Harita Mühendisi Salih BOZKURT, Harita Mühendisi Bahtiyar AKYURT, Harita Mühendisi Fırat ERDEM ve Harita Mühendisi Onur Can BAYRAK' a vermiş oldukları maddi ve manevi desteklerinden dolayı teşekkür ediyorum.

Her zaman sevgi ve sabır gösteren, maddi-manevi destek olan aileme de ayrıca teşekkür ediyorum, şükranlarımı sunuyorum.

Ocak, 2018

Abdulkadir İNCE

## İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	vii
KISALTIMA LİSTESİ.....	viii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ.....	xiii
ÖZET.....	xv
ABSTRACT.....	xviii
<b>BÖLÜM 1</b>	
<b>GİRİŞ</b>	
1.1 Literatür Özeti.....	1
1.2 Tezin Amacı .....	6
1.3 Hipotez .....	7
<b>BÖLÜM 2</b>	
UZAKTAN ALGILAMA VE KIYI ÇİZGİSİ ÇIKARIMI .....	8
2.1 Uzaktan Algılama .....	8
2.2 Kıyı Çizgisi .....	9
2.3 Kıyı Çizgisi Çıkarımında Uzaktan Algılamanın Yeri ve Önemi .....	10
<b>BÖLÜM 3</b>	
YÖNTEMLER .....	12
3.1 Kendini Düzenleyen Haritalar (SOM).....	12
3.1.1 SOM' a Ait Öğrenme Algoritmaları .....	13
3.1.1.1 Sıralı Öğrenme Algoritması.....	14

3.1.1.2	Toplu (Batch) Öğrenme Algoritması .....	15
3.1.2	Toplu Öğrenme ile SOM Algoritması .....	16
3.2	Yapay Sinir Ağları (Artificial Neural Networks – ANN) .....	17
3.2.1	Eğitim Yöntemleri.....	19
3.2.1.1	Gradient Descent (Eğim Azaltımı) Yöntemleri .....	19
3.2.1.2	Conjugate Gradient (Birleşik Eğim) Yöntemler .....	19
3.2.1.2.1	Scaled Conjugate Gradient (Ölçekli Birleşik Eğim) Yöntemi..	20
3.2.1.3	Quasi- Newton (Newton Benzeri) Yöntemleri .....	20
3.2.1.3.1	Levenberg- Marquardt Geri Yayılım Yöntemi.....	21
3.2.2	Yapılarına Göre ANN Modelleri .....	21
3.2.2.1	İleri Beslemeli Ağlar.....	21
3.2.2.1.1	Çok Katmanlı Algılayıcı (Multilayer Perceptron) Ağlar .....	22
3.2.2.1.2	Geriye Yayılım Algoritması.....	24
3.2.2.2	Geri Beslemeli Ağlar .....	28
3.2.3	Öğrenme Türlerine Göre ANN Modelleri .....	29
3.2.3.1	Denetimli Öğrenme .....	29
3.2.3.2	Denetimsiz Öğrenme.....	29
3.2.3.3	Takviyeli Öğrenme.....	29
3.3	Rastgele Orman (RF) .....	30
3.3.1	Sınıflandırma ve Regresyon Ağaçları (CART) .....	30
3.3.1.1	GINI İndeksi.....	31
3.3.2	Rastgele Orman (RF) Algoritması.....	32
3.4	Sayısal Kıyı Çizgisi Analizi Sistemi (Digital Shoreline Analysis System – DSAS) ...	33

## BÖLÜM 4

UYGULAMA .....	36	
4.1	Çalışma Bölgeleri .....	36
4.2	Kullanılan Programlar ve Donanım .....	40
4.3	Uygulamada Kullanılan Veriler ve Özellikleri .....	41
4.4	Uygulama İşlem Adımları .....	42
4.4.1	Kullanılacak Görüntülerin Hazırlanması .....	43
4.4.2	Birinci Aşama .....	46
4.4.2.1	SOM Yöntemine Ait Topoloji Fonksiyonları.....	47
4.4.2.2	SOM Yöntemine Ait Mesafe Fonksiyonları.....	49
4.4.3	İkinci Aşama .....	64
4.4.4	Üçüncü Aşama .....	76

4.4.5 Test Aşaması .....	91
BÖLÜM 5	
SONUÇ VE ÖNERİLER.....	99
KAYNAKLAR.....	107
ÖZGEÇMİŞ.....	112



## SİMGE LİSTESİ

---

$\delta$	Hata terimi
$b_i$	En iyi harita elemanı
$t$	İterasyon sayısı
$\alpha$	Öğrenme oranı
$h_{b_i j}$	Kazanan komşu birimi
$t_{max}$	Maksimum iterasyon
$\mu$	Ölçülerin ortalaması
$\epsilon_i$	Her bir ölçünün hatası
$\sigma$	Standart sapma



## KISALTMA LİSTESİ

---

ANN	Artificial Neural Networks
CART	Classification and Regression Trees
DSAS	Digital Shoreline Analysis System
KA	Karar Ağaçları
NIR	Near Infrared
SOM	Self-organizing Maps
SWIR	Shortwave Infrared
USGS	U. S. Geological Survey

## ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1 Uzaktan algılama platformlarına 2 örnek .....	8
Şekil 2. 2 Kıyı kanununa göre kıyı tanımlarını gösteren kroki [17].....	10
Şekil 3. 1 X ile işaretlenen kazanan birimin ve komşularının konumlarını bu kazanan birime doğru güncellemesi. ....	15
Şekil 3. 2 Yapay zekâ .....	18
Şekil 3. 3 ANN katmanları .....	19
Şekil 3. 4 İleri beslemeli ANN [43] .....	22
Şekil 3. 5 Logistic Sigmoid aktivasyon fonksiyonu [46] .....	23
Şekil 3. 6 Hyperbolic Tangent Sigmoid aktivasyon fonksiyonu [46].....	24
Şekil 3. 7 İleri beslemeli çok katmanlı ANN .....	25
Şekil 3. 8 Geri beslemeli ANN [43] .....	28
Şekil 3. 9 DSAS bileşenleri.....	33
Şekil 3. 10 Hata olasılık fonksiyonu ve standart sapma [59] .....	35
Şekil 4. 1 Terkos çalışma bölgesinin Google Earth üzerindeki görüntüsü .....	37
Şekil 4. 2 İzmir çalışma bölgesinin Google Earth üzerindeki görüntüsü.....	38
Şekil 4. 3 Antalya çalışma bölgesinin Google Earth üzerindeki görüntüsü.....	39
Şekil 4. 4 Erçek Gölü' nün Google Earth üzerindeki görüntüsü .....	40
Şekil 4. 5 Uygulama işlem adımları.....	42
Şekil 4. 6 Uygulamada kullanılan uydu görüntülerine ait NIR-RED-BLUE kombinasyonları .....	44
Şekil 4. 7 Uygulamada kullanılan kırılmış uydu görüntülerine ait NIR-RED-BLUE kombinasyonları.....	45
Şekil 4. 8 Grid ağ topolojisine göre konumlandırılmış nöronlar .....	47
Şekil 4. 9 Altıgen ağ topolojisine göre konumlandırılmış nöronlar .....	48
Şekil 4. 10 Rastgele ağ topolojisine göre konumlandırılmış nöronlar.....	48
Şekil 4. 11 Kombinasyonlara ait ikili sonuç görüntüler .....	50
Şekil 4. 12 Kombinasyonlardaki tek gizli katmanlı ağ tasarımı .....	54
Şekil 4. 13 1 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	54
Şekil 4. 14 2 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	55
Şekil 4. 15 3 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	55

Şekil 4. 16 4 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	56
Şekil 4. 17 5 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	56
Şekil 4. 18 TRAINLM – TANSIG kombinasyonu sonuç görüntüleri (ANN ağı) .....	57
Şekil 4. 19 TRAINLM – LOGSIG kombinasyonu sonuç görüntüleri (ANN ağı) .....	57
Şekil 4. 20 TRAINSCG – TANSIG kombinasyonu sonuç görüntüleri (ANN ağı).....	57
Şekil 4. 21 TRAINSCG – LOGSIG kombinasyonu sonuç görüntüleri (ANN ağı).....	57
Şekil 4. 22 TRAINSCG – TANSIG (Varsayılan) kombinasyonu performans grafiği .....	60
Şekil 4. 23 TRAINSCG – TANSIG (Varsayılan) kombinasyonu gradyan değerleri grafiği..	61
Şekil 4. 24 Sırasıyla 50-250-500 ağaç sayılı yapılara ait sonuç sınıflandırılmış görüntüler (RF ağaç yapıları) .....	62
Şekil 4. 25 50 ağaç sayılı ağaç yapısının 50. Ağacına ait dallanmalar .....	63
Şekil 4. 26 250 ağaç sayılı ağaç yapısının 50. Ağacına ait dallanmalar .....	63
Şekil 4. 27 500 ağaç sayılı ağaç yapısının 50. Ağacına ait dallanmalar .....	63
Şekil 4. 28 SOM yöntemi sonuçları (1. sütun SOM yöntemi sonuçları, 2. sütun kara kümeleri, 3. sütun deniz kümeleri) .....	65
Şekil 4. 29 1 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	67
Şekil 4. 30 2 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	67
Şekil 4. 31 3 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	68
Şekil 4. 32 4 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	68
Şekil 4. 33 5 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz) .....	69
Şekil 4. 34 TRAINLM – TANSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı) .....	69
Şekil 4. 35 TRAINLM – LOGSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı) .....	70
Şekil 4. 36 TRAINSCG – TANSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı) .....	70
Şekil 4. 37 TRAINSCG – LOGSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı) .....	70
Şekil 4. 38 TRAINSCG – TANSIG (Sadece Eğitim) kombinasyonu performans grafiği .....	73
Şekil 4. 39 TRAINSCG – TANSIG (Sadece Eğitim) kombinasyonu gradyan değerleri grafiği .....	73
Şekil 4. 40 Sırasıyla 50-250-500 ağaç sayılı yapılara ait sonuç sınıflandırılmış görüntüler (SOM – RF ağaç yapıları) .....	74
Şekil 4. 41 Birinci ve ikinci aşamada en iyi sonucu veren kıyı çizgileri ve DSAS bileşenleri .....	75
Şekil 4. 42 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan) .....	77
Şekil 4. 43 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan) .....	77
Şekil 4. 44 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan) .....	78
Şekil 4. 45 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan) .....	78

Şekil 4. 46 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim) .....	78
Şekil 4. 47 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim) .....	78
Şekil 4. 48 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim) .....	78
Şekil 4. 49 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim) .....	79
Şekil 4. 50 1000 iterasyon - 10 gizli katman (Varsayılan) kombinasyonu performans grafiği .....	83
Şekil 4. 51 1000 iterasyon - 10 gizli katman (Varsayılan) kombinasyonu gradyan değerleri grafiği .....	84
Şekil 4. 52 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan) .....	84
Şekil 4. 53 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan) .....	84
Şekil 4. 54 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan) .....	84
Şekil 4. 55 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan) .....	85
Şekil 4. 56 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim) .....	85
Şekil 4. 57 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim) .....	85
Şekil 4. 58 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim) .....	85
Şekil 4. 59 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim) .....	85
Şekil 4. 60 1000 iterasyon - 5 gizli katman (Varsayılan) kombinasyonu performans grafiği .....	89
Şekil 4. 61 1000 iterasyon - 5 gizli katman (Varsayılan) kombinasyonu gradyan değerleri grafiği .....	90
Şekil 4. 62 Üçüncü aşamada en iyi sonucu veren kıyı çizgileri ve DSAS bileşenleri .....	90
Şekil 4. 63 Test aşamasında kullanılan ANN bileşenlerin elde edildiği aşamaların gösterimi .....	91
Şekil 4. 64 7 numaralı test görüntüsü ve elde edilen sonucu .....	92
Şekil 4. 65 8 numaralı test görüntüsü ve elde edilen sonucu .....	92
Şekil 4. 66 9 numaralı test görüntüsü ve elde edilen sonucu .....	92
Şekil 4. 67 10 numaralı test görüntüsü ve elde edilen sonucu .....	93
Şekil 4. 68 11 numaralı test görüntüsü ve elde edilen sonucu .....	93
Şekil 4. 69 12 numaralı test görüntüsü ve elde edilen sonucu .....	93
Şekil 4. 70 Test aşamasında kullanılan Terkos bölgesine ait kıyı çizgileri ve DSAS bileşenleri .....	94
Şekil 4. 71 Test aşamasında kullanılan İzmir bölgesine ait kıyı çizgisi ve DSAS bileşenleri .....	95

Şekil 4. 72 Test aşamasında kullanılan Antalya bölgesine ait kıyı çizgisi ve DSAS bileşenleri.....	96
Şekil 4. 73 Test aşamasında kullanılan Erçek Gölü' ne ait kıyı çizgisi ve DSAS bileşenleri .....	97
Şekil 5. 1 Birinci ve ikinci aşamaya ait yöntemlerin en iyi DSAS sonuçları .....	100
Şekil 5. 2 Birinci ve ikinci aşamaya ait en iyi sonuçların elde edilme süreleri .....	101
Şekil 5. 3 Üçüncü aşamaya ait en iyi DSAS sonuçları.....	103
Şekil 5. 4 Üçüncü aşamaya ait en iyi sonuçların elde edilme süreleri.....	103
Şekil 5. 5 Test aşamasına ait DSAS sonuçları .....	105



## ÇİZELGE LİSTESİ

	Sayfa
Çizelge 4. 1 LANDSAT 8 uydusuna ait bantların spektral aralıkları ve çözünürlükleri ....	41
Çizelge 4. 2 Uygulamada kullanılan uydu görüntülerinin tarihleri .....	42
Çizelge 4. 3 Kullanılan görüntülerin satır ve sütun sayıları .....	46
Çizelge 4. 4 SOM parametreleri .....	47
Çizelge 4. 5 SOM kombinasyonları .....	50
Çizelge 4. 6 Kombinasyon sonuçlarının elde edilme süreleri .....	51
Çizelge 4. 7 Kombinasyonlara ait kara ve deniz piksel sayıları .....	52
Çizelge 4. 8 Tasarlanan ağ kombinasyonları .....	53
Çizelge 4. 9 ANN ağlarının eğitime süreleri .....	58
Çizelge 4. 10 ANN ağlarının ortalama karesel hataları .....	58
Çizelge 4. 11 ANN ağlarının gradyan değerleri .....	58
Çizelge 4. 12 ANN ağlarına ait karşılaştırma sonuçlarının ortalama hataları .....	59
Çizelge 4. 13 Görüntülerin elde edilme süreleri .....	62
Çizelge 4. 14 Sonuç görüntülere ait karşılaştırma sonuçları .....	64
Çizelge 4. 15 Karadeniz bölgesine ait görüntülerin elde edilme süreleri .....	66
Çizelge 4. 16 SOM – ANN ağlarının eğitime süreleri .....	70
Çizelge 4. 17 SOM – ANN ağlarının ortalama karesel hataları .....	71
Çizelge 4. 18 SOM – ANN ağlarının gradyan değerleri .....	71
Çizelge 4. 19 SOM – ANN ağlarına ait karşılaştırma sonuçlarının ortalama hataları .....	72
Çizelge 4. 20 Görüntülerin elde edilme süreleri .....	74
Çizelge 4. 21 Sonuç görüntülere ait karşılaştırma sonuçları .....	75
Çizelge 4. 22 Birinci ve ikinci aşamada en iyi sonuç veren kıyı çizgilerinin ortalama hataları (DSAS sonuçları) .....	76
Çizelge 4. 23 Gizli katman ve iterasyon sayıları ile oluşturulan kombinasyonlar .....	77
Çizelge 4. 24 ANN-3 ağlarının eğitime süreleri .....	79
Çizelge 4. 25 ANN-3 ağlarının ortalama karesel hataları .....	80
Çizelge 4. 26 ANN-3 ağlarının gradyan değerleri .....	81
Çizelge 4. 27 ANN-3 ağlarına ait karşılaştırma sonuçlarının ortalama hataları .....	82
Çizelge 4. 28 SOM – ANN-3 ağlarının eğitime süreleri .....	86
Çizelge 4. 29 SOM – ANN-3 ağlarının ortalama karesel hataları .....	86
Çizelge 4. 30 SOM – ANN-3 ağlarının gradyan değerleri .....	87
Çizelge 4. 31 SOM – ANN-3 ağlarına ait karşılaştırma sonuçlarının ortalama hataları .....	88

Çizelge 4. 32 Üçüncü aşamada en iyi sonuç veren kıyı çizgilerinin ortalama hataları (DSAS sonuçları) .....	91
Çizelge 4. 33 Test aşamasına ait kıyı çizgilerinin ortalama hataları (DSAS sonuçları).....	98
Çizelge 5. 1 Şekil 5. 1' deki kıyı çizgilerinin hatasız hale getirilmesi için harcanan süreler .....	99
Çizelge 5. 2 Şekil 5. 3' deki kıyı çizgilerinin hatasız hale getirilmesi için harcanan süreler .....	101
Çizelge 5. 3 Test görüntülerinden elle yapılan sayısallaştırmalar için harcanan zamanlar .....	103
Çizelge 5. 4 Test sonucu elde edilen kıyı çizgilerinin hatasız hale getirilmesi için harcanan süreler .....	103



## YAPAY SİNİR AĞLARI VE RASTGELE ORMAN YÖNTEMLERİ İLE LANDSAT 8 GÖRÜNTÜLERİNDEN OTOMATİK KIYI ÇİZGİSİ ÇIKARTILMASI

Abdulkadir İNCE

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Bülent BAYRAM

İnsanların suya olan vazgeçilmez ihtiyacı, kıyı alanlarını insanların en çok kullandığı yerleşim yerleri haline getirmiştir. Kıyılar doğal habitat için en uygun habitatlar oluşturmaktadır. Kıyılar aynı zamanda mikroklimatik özellikleri sayesinde tatil ve dinlenme olanakları sağlayan alanlardır. Turizm, sanayi, kültür balıkçılığı, kentleşme gibi faaliyetler kıyı alanları açısından en büyük tehditlerden bazılarıdır. Bu tehditler, kıyı alanlarının korunması ve sürdürülebilir kıyı yönetimi için kıyı alanlarının izlenmesini zorunlu hale getirmiştir. Doğal çevre yönetimi, afet yönetimi, kıyı erozyonu incelemeleri, katı madde taşınımı ve kıyı morfolojilerinin modellenmesi gibi farklı alanlarda kıyı çizgileri yaygın olarak kullanılmaktadır. Bu sebeple kıyı çizgilerinin özellikle uydu görüntülerinden elde edilmesi için çeşitli yöntemler kullanılarak birçok çalışma gerçekleştirilmiştir. Son yıllarda yapay zekâ kullanılarak elde edilen başarılar, çeşitli alanlarda çalışan bilim insanlarını da yaptıkları çalışmalarda yapay zekâ kullanmaya teşvik edici olmuştur.

Sunulan tez çalışması kapsamında, uydu görüntülerinden kıyı çizgisi çıkarımı kendini düzenleyen haritalar, yapay sinir ağları ve rastgele orman yöntemleri kullanılarak gerçekleştirilmiştir.



Yürütücülüğünü Prof. Dr. Bülent Bayram' ın yaptığı 115Y718 numaralı ve "SÜRDÜRÜLEBİLİR KIYI ALANI İZLEME MODELİ İÇİN İHA VE İHA-LİDAR VERİLERİNDEN OTOMATİK ÜÇ BOYUTLU KIYI ÇİZGİSİ ÇIKARTILMASI VE ANALİZİ: TERKOS (İSTANBUL) ÖRNEĞİ" başlıklı TÜBİTAK projesinin test alanı olan İstanbul'un Çatalca ilçesine bağlı Terkos bölgesinin Karadeniz kıyısı yöntemlerin test edilmesi için seçilmiştir. Tez kapsamında önerilen yöntemlerin söz konusu proje açısından uygulanabilirliği test edilmiştir.

Karadeniz Bölgesi' nin 5 farklı konumuna ait 2017 yılına ait LANDSAT 8 görüntüsü eğitim amaçlı olarak kullanılmıştır. İstanbul-Terkos bölgesine ait 2013, 2015, 2017 tarihlerine ait aynı mevsimde elde edilmiş üç veri seti, Antalya bölgesine ait 2017 yılında elde edilmiş bir LANDSAT 8, İzmir bölgesine ait 2017 yılında elde edilmiş bir LANDSAT 8, ve Erçek Gölü' ne ait 2018 yılında elde edilmiş bir görüntü olmak üzere toplam altı farklı LANDSAT 8 görüntüsü test amaçlı olarak kullanılmıştır.

Uygulamanın birinci aşamasında Kendini Düzenleyen Haritalar Yöntemi için Terkos bölgesine ait uydu görüntüsü kara ile deniz olacak şekilde kümelenecek ve kıyı çizgisi çıkartılmıştır. Yapay Sinir Ağları Yöntemi için Karadeniz Bölgesi' ne ait 5 farklı uydu görüntüsü kullanılarak karadan ve denizden 2 adet sınıf oluşacak şekilde örnek veriler toplanmıştır. Yapay sinir ağları bu veriler ile eğitilmiş ve ağların test edilmesi için Terkos bölgesine ait uydu görüntüsü eğitilen ağlar ile sınıflandırılarak kıyı çizgileri çıkartılmıştır. Rastgele Orman Yönteminde ağaç yapılarını oluşturmak için yapay sinir ağlarının eğitiminde kullanılan eğitim verileri kullanılmıştır.

Bu tezin ana amacı, Yapay Sinir Ağları Yönteminde kullanılan eğitim verileri ile Rastgele Orman Yönteminde ağaç yapılarını oluşturmak için kullanılan verilerinin Kendini Düzenleyen Haritalar Yöntemi kullanılarak otomatik olarak toplanmasını sağlamaktır.

İkinci aşamada Karadeniz Bölgesi' ne ait uydu görüntüleri Kendini Düzenleyen Haritalar Yöntemi kullanılarak kara ile deniz olacak şekilde kümelenecektir. Kümeleme sonuçlarına göre orijinal görüntülerden deniz ve kara sınıfları ayrılmıştır. Daha sonra, elde edilen bu veriden yararlanılarak yapay sinir ağları için her bir görüntüye ait deniz ve kara kümelerinden rastgele örnek veriler toplanmıştır. Rastgele toplanan bu veriler ile yapay sinir ağları eğitilmiş ve test alanına ait görüntülerden kıyı çizgileri çıkartılmıştır. Rastgele Orman Yöntemi için ise yapay sinir ağlarında kullanılan rastgele toplanmış veri seti, ağaç yapılarını oluşturmak için kullanılmıştır. Oluşturulan bu ağaç yapılarına göre test bölgesine ait uydu görüntüleri sınıflandırılmış ve kıyı çizgileri çıkartılmıştır.

İlk iki aşamada kullanılan yapay sinir ağları Levenber – Marquardt (TRAINLM), Ölçekli Conjugate Gradient (TRAINSCG) eğitim fonksiyonları ile Hyperbolic Tangent Sigmoid (TANSIG), Logistic Sigmoid (LOGSIG) bağlantı fonksiyonlarının kombinasyonlarından tek gizli katmanlı ve 2000 iterasyonlu olarak tasarlanmıştır.

Son aşama olan üçüncü aşamada TRAINSCG – TANSIG kombinasyonu kullanılarak farklı sayıda gizli katman (5, 10, 15) ve iterasyon sayılarıyla (500, 1000, 1500, 2000) yeni ağlar tasarlanmıştır. Gizli katman ve iterasyon sayılarının oluşturdukları kombinasyonlar ile tasarlanan ağlar hem kullanıcı tarafından toplanan eğitim verileri ile hem de Kendini Düzenleyen Haritalar Yöntemi sonuçlarından rastgele toplanan eğitim verileri ile eğitilerek kıyı çizgileri elde edilmiştir.

Öncelikle belirlenen kombinasyonların test edilmesi amacıyla önerilen yöntemler İstanbul-Terkos, 2017 Landsat 8 görüntüsüne uygulanmıştır. Birinci aşamadaki Kendini Düzenleyen Haritalar Yöntemi ile elde edilen kıyı çizgisi elle yapılan sayısallaştırma sonuçları ile karşılaştırılmış ve ortalama 0,49 piksel hata ile kıyı çizgisi çıkarılmıştır. Kullanıcı tarafından toplanan eğitim verileri ile eğitilen yapay sinir ağları içerisinde TRAINSCG – TANSIG kombinasyonu ortalama 0,61 piksel hata ile en iyi sonucu vermiştir. Aynı eğitim verilerinin kullanıldığı Rastgele Orman yöntemi sonuçları içerisinde ise en iyi sonucu 50 ağaçtan oluşan ve ortalama 0,45 piksel hata ile kıyı çizgisi elde edilmiştir. İkinci aşamadaki Kendini Düzenleyen Haritalar Yöntemi kümeleme sonuçlarına göre rastgele toplanan eğitim verileri kullanılarak eğitilen ağlardan elde edilen en iyi sonuç ortalama 0,22 piksel hata ile TRAINSCG – TANSIG kombinasyonuna aittir. Aynı rastgele eğitim verileri kullanılarak oluşturulan ağ yapısı içerisinde 250 ağaçlı ağ yapısı ortalama 0,79 piksel hata ile kıyı çizgisini tespit etmiştir. Üçüncü aşamadaki kombinasyonlarda kullanıcı tarafından toplanan eğitim verileri kullanılarak oluşturulan ağlardan en iyi sonucu ortalama 0,36 piksel hata ile 10 gizli katmanlı ve 1000 iterasyonlu yapay sinir ağı vermiştir. Kendini Düzenleyen Haritalar Yöntemi kümeleme sonuçlarına göre rastgele toplanan eğitim verileri kullanılarak eğitilen ağlardan elde edilen en iyi sonuç ise ortalama 0,20 piksel hata ile 5 gizli katmanlı ve 1000 iterasyonlu yapay sinir ağına aittir.

Bu aşamanın ardından seçilen en iyi konfigürasyonlar göz önünde bulundurularak 6 ek test verisi kullanılmıştır. Elle sayısallaştırma sonuçları ile elde edilen sonuçlar karşılaştırılmıştır. İstanbul-Terkos bölgesi 2013 yılına ait LANDSAT 8 görüntüsünden ortalama 0,47 piksel, İstanbul-Terkos bölgesi 2015 yılına ait LANDSAT 8 görüntüsünden ortalama 0,36 piksel, İstanbul-Terkos bölgesi 2017 yılına ait LANDSAT 8 görüntüsünden ortalama 0,36 piksel, Erçek Gölü'nün 2017 yılına ait LANDSAT 8 görüntüsünden ortalama 0,57 piksel, İzmir bölgesinin 2017 yılına ait LANDSAT 8 görüntüsünden ortalama 0,31 piksel ve Antalya bölgesinin 2017 yılına ait LANDSAT 8 görüntüsünden ortalama 1,31 piksel hata ile kıyı çizgileri tespit edilmiştir.

**Anahtar Kelimeler:** Kendini düzenleyen haritalar, yapay sinir ağları, rastgele orman, görüntü işleme, uzaktan algılama, kıyı çizgisi çıkarımı

**AUTOMATIC SHORELINE EXTRACTION FROM LANDSAT 8 IMAGERIES  
WITH ARTIFICIAL NEURAL NETWORKS AND RANDOM FOREST METHODS**

Abdulkadir İNCE

Department of Geomatics Engineering

MSc. Thesis

Adviser: Prof. Dr. Bülent BAYRAM

Coastal areas are always very important for human being because of their habitat. In addition to it, coastal areas provide vacation and relaxation opportunities because of their microclimatic features. Activities such as tourism, industry, aquaculture and urbanization are main threads for coastal areas. Therefore, monitoring of coastal areas one of the vital issues for preserving and sustainable management of environmental heritage. Shorelines are widely used in different areas such as natural environment management, disaster management, coastal erosion studies, solid matter transport and modeling of coastal morphodynamics. For this reason, many studies have been carried out using various methods in order to extract the shoreline from satellite images in particular. Due to their advantages, Artificial Neural Network (ANN) based methods became very popular for many scientists.

In this thesis, shoreline extraction from satellite image was performed by using SOM, ANN and Random Forest (RF) methods. In this study, Terkos region has been chosen for testing of the proposed methods within the scope of "TUBITAK Project (Project No: 115Y718) titled" Integration of Unmanned Aerial Vehicles for Sustainable Coastal Zone Monitoring Model – Three-Dimensional Automatic Coastline Extraction and Analysis: Istanbul-Terkos Example “.

Landsat 8 imageries have been used to implement proposed algorithms. 5 Landsat 8 images have been used for training which were from different part of Black Sea region of Turkey and taken in the year of 2017. 6 Landsat 8 images were used for testing of proposed methods First data set is from Terkos-Istanbul in the years of 2013, 2015, 2017. Second one is from province of Antalya which was taken in the year of 2017, third one is from province of İzmir which was taken in the year of 2017 and last data set is from Lake Ercek which was taken in the year of 2017. NIR-RED-BLUE bands of all test data have been used for this thesis.

In the first phase of the thesis, using SOM method, satellite image of Terkos region is clustered as two clusters (land – sea) and shoreline has been extracted from the clustered image. To train the ANN method, 5 different satellite images belonging to the Black Sea Region were used. From these images, sample data were collected to form 2 classes (land – sea). The same training areas have been used for both ANN and RF methods.

The main aim of this thesis is to collect automatically using results of SOM method and to train ANN model and to construct tree structures for the RF method.

In the second phase of the thesis, the training data-set have been classified to obtain land and sea classes. Following to this step, training of artificial network and generating of the tree structure was carried out using randomly selected data from the results of SOM. After training, SOM, ANN and RF methods have been implemented to the test images.

The artificial neural networks used in the first and second steps were consisted of single hidden layer and 2000 iterations by combination of Levenber-Marquardt (TRAINLM) and Scale Conjugate Gradient (TRAINSCLG) training functions with Hyperbolic Tangent Sigmoid (TANSIG) and Logistic Sigmoid (LOGSIG) transfer functions.

In the final step, new networks were designed using the TRAINSCLG - TANSIG combination with different numbers of hidden layers (5, 10, 15) and iteration numbers (500, 1000, 1500, 2000). The manually selected training data and results of SOM method has been used separately as training data set for all combinations.

First of all, to test the defined combinations of the proposed methods were applied on Istanbul-Terkos, 2017 Landsat 8 image. The shoreline obtained by the SOM Method in the first step was compared with the manual digitization and the average error was calculated as 0,49 pixels. Among the artificial neural networks trained by user - collected training data, the TRAINSCLG - TANSIG combination gave the best result with an average error of 0,61 pixels. In the results of the RF method using the same training data, which consists of 50 trees, average shoreline extraction error was 0,45 pixels.

Based on the clustering results of the SOM Method at the second step, shoreline extraction error was calculated as 0,22 pixels by using TRAINSCLG - TANSIG combination. The same training data were used for RF method and created 250 trees. As a result, average error was calculated as 0,79 pixels.

In the third step, the best resultant network from the user-collected training data provided 10 hidden layers and 1000 iterative artificial neural networks was used and average shoreline extraction error was calculated as 0,36 pixels. The use of SOM results and by selecting of randomly training data, the ANN configuration with 5 hidden layers and 1000 iteration, the shoreline extraction error was calculated as 0.20 pixels.

After this step, by considering of best combinations of proposed methods, using additional 6 images, methods were tested again. For each image accuracy assessment was realized by comparison of manual digitizing results with obtained results.

The shoreline was extracted with an average error of 0,47 pixels from the LANDSAT 8 image of the Terkos region in the year of 2013. In the same way the shoreline extracted with an average error of 0,36 pixels from the LANDSAT 8 image of the Terkos region in the year of 2015. For Istanbul-Terkos, 2017 Landsat 8 image, average extraction error was calculated as 0.22 pixels. The shorelines were extracted with an average error of 0.57 pixels from the LANDSAT 8 image of Ercek Lake in the year of 2017, an average error of 0.31 pixels from the LANDSAT 8 image of the Izmir region in the year of 2017 and an average error of 1.31 pixels from the LANDSAT 8 image of the Antalya region in the year of 2017.

**Keywords:** Self-Organizing Maps, artificial neural networks, random forest, image processing, remote sensing, shoreline extraction

#### 1.1 Literatür Özeti

Awad (2010) tarafından yapılan çalışmada, mevcut olan birçok görüntü bölütleme yöntemlerinin uydu görüntüleri için uygun olmadığını ve bu tekniklerin uygulanabilmesi için önceden bir bilgiye ihtiyaç duyulduğundan bahsedilmiştir. Bu engellerin üstesinden gelmek için denetimsiz yapay sinir ağı metodu olan kendini düzenleyen haritalar (SOM) ve eşik değer metodunu kullanarak, TSOM şeklinde tanımlanan yeni bir uydu görüntüsü bölütleme yöntemi geliştirilmiştir. TSOM yöntemi ile ISODATA (iteratif olarak kendini düzenleyen veri) yöntemini farklı çözünürlüklere sahip 2 uydu görüntüsü kullanarak karşılaştırılmış ve TSOM yönteminin ISODATA yöntemine göre daha doğru sonuçlar verdiği gözlemlenmiştir [1].

Yuan ve diğ. (2009) tarafından yapılan çalışmada, otomatik bir yapay sinir ağı sınıflandırma sistemi üzerinde durulmuştur. Yapay sinir ağları (Artificial Neural Networks – ANN), SOM yöntemi ve SOM yönteminin benzetilmiş tavlama (simulated annealing) yöntemini ile birleştirildiği bir yöntem sunulmuştur. Bu yöntemler kullanılarak LANDSAT uydu görüntüsü sınıflandırılmış ve aynı bölgeye ait yüksek çözünürlüklü ortofoto haritadan elde edilen sınıf verileri ile karşılaştırılarak sonuçlar sunulmuştur. Diğer 2 yonteme kıyasla ANN yönteminin, daha doğru sonuçlar verdiği gözlemlenmiştir. Ayrıca, SOM yönteminin benzetilmiş tavlama (simulated annealing) yöntemi ile birleştirildiği yöntem, SOM yöntemine göre daha iyi sonuç verdiği gözlemlenmiştir [2].

Marandi ve diğ. (2017) gerçekleştirdikleri çalışmada, sınıflandırma işlemlerinin bilim, endüstri, işletme ve sağlık sistemlerin ayrılmaz bir parçası olduğunu ve sınıflandırma işlemlerinin bu kadar yaygın olmasından dolayı en küçük bir gelişmenin bile değerli olduğunu vurgulamaktadır. Birçok disiplinde sınıflandırma işlemlerinde kullanılan en güçlü yöntemlerden biri olan ANN ile SOM yöntemini birleştirerek, kendiliğinden organize hata odaklı yapay sinir ağı adını verdikleri metot ile klasik ANN yöntemini 5 farklı veri seti kullanarak karşılaştırılmıştır. Önerilen yöntemin daha doğru ve güvenilir olduğunu ortaya koyduğu gözlemlenmiştir [3].

Akar ve Güngör (2013) tarafından yapılan çalışmada, farklı konumsal çözünürlüğe ve karakteristiğe sahip çok bantlı uydu görüntüleri kullanarak rastgele orman (RF) yönteminin performansı incelenmiştir. 2005 ve 2008 yıllarına ait Ikonos ve QuickBird uydu görüntülerinin kullanıldığı bu çalışmada, sınıflandırma sonuçları Gentle AdaBoost (GAB), En Çok Benzerlik (EÇB) ve Destek Vektör Makineleri (DVM) yöntemlerinden elde edilen sonuçlarla karşılaştırılmıştır. Elde edilen sonuçlarda, RF yönteminin daha iyi sonuçlar ortaya koyduğunu gözlemlenmiştir [4].

Pal (2013) tarafından gerçekleştirilen çalışmada, RF ve destek vektör makineleri (DVM) yöntemlerinin sınıflandırma sonuçlarını, sınıflandırma doğruluğuna, eğitim süresine ve kullanıcı tarafından belirlenen parametrelere göre karşılaştırmıştır. LANDSAT uydu görüntüsünün kullanıldığı bu çalışmanın sonuçlarında, sınıflandırma doğruluğu ve eğitim süresi bakımından RF yönteminin, DVM yöntemine eşit derecede iyi performans gösterdiği gözlemlenmiştir. Ayrıca RF yönteminin, DVM yöntemine göre kullanıcı tarafından belirlenen parametre sayısının daha az olduğu vurgulanmıştır [5].

Bayram ve diğ. (2017) yapmış oldukları çalışmada, Terkos bölgesine ait farklı çözünürlüklere sahip LANDSAT ve GÖKTÜRK uydu görüntüleri kullanılmıştır. RF yönteminin yüksek ve düşük çözünürlüklü görüntülerde performansı ölçülmüştür. Elle yapılan sayısallaştırmaya ile yapılan doğruluk değerlendirmelerine göre, RF yönteminin hem yüksek çözünürlüklü hem de düşük çözünürlüklü görüntülerde iyi sonuçlar ortaya koyduğu gözlemlenmiştir [6].

Bayram ve diğ. (2017) tarafından yapılan çalışmada, sayısal ortofotolardan kıyı çizgisi çıkarımında 2 farklı yalınlaştırma yöntemi, yalınlaştırma yapılmadan elde edilen kıyı çizgisi ile karşılaştırılmıştır. Letonya' nın Riga bölgesine ve 2013 yılına ait 7 adet ortofotonun Near-Infrared (NIR) bantlarına ait görüntülerden Yalın Doğrusal Yinelemeli Kümeleme (SLIC) yöntemiyle kıyı çizgileri üretilmiştir. Bend ve Douglas-Peucker yalınlaştırma yöntemleriyle elde edilen kıyı çizgileri yalınlaştırma yapılmadan elde edilen SLIC yöntemi sonucu ile karşılaştırılmıştır. Yapılan çalışma sonucunda yalınlaştırma yöntemlerinden elde edilen kıyı çizgilerinin birbirine daha yakın ve doğru sonuçlar verdiği görülmüştür [7].

Reis ve Bayram (2016) tarafından gerçekleştirilen çalışmada, Parçacık Sürü Optimizasyonu (PSO) yöntemi ile Erçek Gölü'nün 2013 ve 2014 yıllarına ait RASAT uydu görüntülerinden kıyı çizgileri çıkartılmıştır. PSO yöntemi ile elde edilen kıyı çizgileri elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Yapılan çalışma sonucunda kullanılan yöntemin diğer yöntemlere göre daha iyi performans gösterdiği görülmüştür [8].

Aydın ve UANNI (2013) gerçekleştirdikleri çalışmada, Sakarya ilinin Karasu ilçesinin kıyı değişimi 1987, 1995, 2003, 2011 ve 2013 yıllarına ait LANDSAT uydu görüntülerinden elle sayısallaştırma yapılarak yıllara göre değişim gözlemlenmiştir. Elde edilen sonuçlara göre Sakarya Nehrinin Karadeniz' e döküldüğü nehir ağzının her iki tarafında 1987 yılına göre 300 m' ye varan ciddi bir değişim gözlemlenmiştir [9].

Kuleli (2010) tarafından yapılan çalışmada, 1989, 1999 ve 2009 yıllarına ait LANDSAT TM uydu görüntüleri kullanılarak Kızılırmak ve Yeşilirmak nehir ağzlarında yıllara göre kıyı çizgisi değişim oranları hesaplanmıştır. 20 yıllık bir değişimde Kızılırmak nehir kıyı çizgisininin 660 m, Yeşilirmak nehir kıyı çizgisininin ise 620 m' ye varan oranda kara tarafında ilerlediği belirlenmiştir [10].



Dođan (2008) tarafından gerekleřtirilen yksek lisans tezinde, 1963 yılına ait Corona, 1987 ile 2000 yıllarına ait LANDSAT ve 2007 yılına ait Aster uydu grntleri kullanılarak Alaatı blgesindeki kıyı izgisi ve kıyı alanı kullanımındaki zamansal deđiřim belirlenmiřtir. 1963 yılından 2007 yılına kadar toplamda 8,1119 hektar alanın denizin doldurulmasıyla kazanıldıđı tespit edilmiřtir. 1963 yılı ile 2007 yılları arası kıyı izgisi deđiřiminin ise 42 m ile 132 m arasında deđiřtiđi belirlenmiřtir [11].

Upadhyay ve Chhabra (2015) tarafından yapılanı alıřmada, deđiřtirilmiř bir SOM yapay sinir ađı (MSOM ANN) grnt bltleme yntemi sunulmuřtur. zellik ıkarma ve eđitim veri setinin olmadıđı nerilen yntemde, grntler gri deđer yođunluklarına gre sınıflandırılmıřtır. Klasik SOM yapay sinir ađına (SOM ANN) fazladan bir katman eklenerek SOM ANN sonucu oluřan kmelerin gri deđerleri, girdi verilerin gri deđerlerine gre belirlenmiřtir. nerilen bu yntemin test edilmesi iin Berkeley rneklemeye Veri Setini kullanılmıř ve elde edilen sonular Bulanık C-Ortalamalar yntemi ile elde edilen sonularla karřılařtırılmıřtır. nerilen MSOM ANN ynteminin Bulanık C-Ortalamalar ynteminden daha iyi sonular ortaya koyduđu tespit edilmiřtir [12].

Pascula ve diđ. (2012) tarafından yapılan alıřmada, nerilen alt piksel otomatik kıyı izgisi ıkarımı algoritması ile ardıřık zamanlı uydu grntlerinden otomatik kıyı izgisi ıkarımı gerekleřtirilmiřtir. 30 m znrlkl LANDSAT TM (28 adet) ve LANDSAT ETM+ (10 adet yksek kazanımlı, 7 adet dřk kazanımlı) grntleri kullanılmıřtır. Elde edilen kıyı izgilerinin cođrafi koordinatlarının dođruluklarını arttırmak iin alıřılan blgeye ait 0,5 m znrlkl ortofoto harita kullanılmıřtır. Elde edilen kıyı harita zerinde yapılmıř olan kıyı izgisi ile karřılařtırıldıđında, yksek kazanımlı LANDSAT ETM+ grntlerinde maksimum 10,54 m hata, LANDSAT TM grntlerinde maksimum 11,09 m hata ve dřk kazanımlı LANDSAT ETM+ grntlerinde maksimum 13,24 m hata elde edilmiřtir [13].

Acar ve diğ. (2012) tarafından gerçekleştirilen çalışmada, SAR uydu görüntülerinden otomatik kıyı çizgisi çıkarımı yapan bir algoritma geliştirilmiştir. Zonguldak ve Bartın illerini kapsayan 2007 ve 2010 yıllarına ait HH ve HV polarizasyonlu toplam 4 PALSAR uydu görüntüsünden kıyı çizgileri çıkarılmış ve elle yapılan sayısallaştırma sonuçlarıyla karşılaştırılmıştır. Elde edilen kıyı çizgileri uzunluklarına göre karşılaştırıldığında, 2007 yılına ait HV polarizasyonlu görüntüden yüzde 0,61 ve HH polarizasyonlu görüntüden yüzde 0,85 doğrulukla kıyı çizgileri elde edilmiştir. 2010 yılına ait HV polarizasyonlu görüntüden yüzde 0,41 ve HH polarizasyonlu görüntüden yüzde 0,21 doğrulukla kıyı çizgileri elde edilmiştir [14].

Olgun (2012) tarafından yapılan yüksek lisans tezinde, Göksu deltasının 1980-2008 yılları arası kıyı değişimi gözlemlenmiş ve analiz edilmiştir. 1/25000 ve 1/100000 ölçekli topoğrafik haritalar, LANDSAT TM ve TERRA/ASTER uydu görüntüleri ile Yüksek Su Seviyesi (YSS)' den alınan GPS (Global Positioning System) ölçümleri kullanılmıştır. Yapılan çalışma sonucunda, kıyı değişimi ve hesaplanan yıllık değişim oranlarının delta gelişimine paralel olduğu ve bu değişimin nehirler tarafından taşınan malzeme miktarı ile orantılı olduğu sonucuna ulaşılmıştır [15].

Kalkan ve diğ. (2010) tarafından gerçekleştirilen çalışmada, 2002 yılına ait LANDSAT ETM+ uydu görüntüsünden otomatik kıyı çizgisi çıkarımı yapılarak, bu kıyı çizgisi Coğrafi Bilgi Sistemi (CBS) platformlarında kullanılabilir hale getirilmiştir. Beyşehir gölüne ait uydu görüntüsü, nesne tabanlı görüntü işleme yazılımı olan eCognition Developer ile segmente edilmiş ve daha sonra MATLAB ortamında eşikleme yöntemiyle görüntüdeki göl sınıf seçilerek CBS platformlarında kullanılacak uzantılı olarak kıyı çizgisi çıkarılmıştır [16].

Sesli (2006) tarafından yapılan çalışmada, Trabzon iline ait Orman Genel Müdürlüğü'ne bağlı Trabzon Bölge Müdürlüğü'nden temin edilen 1973 yılına ait 1/23000 ölçekli hava fotoğrafları ve 2002 yılına ait 1/15000 ölçekli kızılötesi hava fotoğrafları kullanılarak kıyı çizgisi ve kıyı bölgesi kullanımındaki değişim belirlenmiştir. Çalışma sonucunda, kıyı çizgisindeki zamansal değişimler saptanarak dolgu yapılan alanlar ile kıyı alanı kayıplarının büyüklükleri tespit edilmiştir. 1973-2002 yılları arasında 16 farklı yerde toplam 101 hektarlık alanın doldurularak kazanıldığı ve 19 farklı yerde toplam 11,3 hektarlık alanın kaybedildiği sonucuna ulaşılmıştır [17].

İnce ve diğ. (2017) tarafından yapılan çalışmada, Sapanca Gölü'ne ait 2012 tarihli RASAT uydu görüntüsünden nesne tabanlı Meanshift ve SLIC yöntemleri kullanarak otomatik kıyı çizgisi çıkarımı gerçekleştirilmiştir. Kullanılan yöntemler sonucu elde edilen kıyı çizgilerini elle yapılan sayısallaştırma sonucu ile karşılaştırılmıştır. Meanshift yönteminin alansal doğruluğu 0,27 km<sup>2</sup> ve konumsal doğruluğu 7,66 m, SLIC yöntemi için alansal doğruluğu 0,06 km<sup>2</sup>, konumsal doğruluğu ise 6,07 m bulunmuştur [18].

## **1.2 Tezin Amacı**

Sunulan bu tezin amacı, uydu görüntülerinden kıyı çizgisi çıkarımı için ANN ve RF yöntemlerinde kullanıcı tarafından seçilen eğitim verilerinin SOM yöntemini kullanarak otomatik olarak elde edilmesidir. SOM yöntemi kullanılarak tasarlanan ANN ve ağaç yapıları ile kullanıcı kaynaklı hataların elimine edilmesi ve zaman konusunda tasarruf sağlanması amaçlanmıştır.

### 1.3 Hipotez

Sunulan bu tezde,

- “SOM yönteminin kıyı çizgisi çıkarımında kullanılması uygun mudur?”
- “Farklı uydu görüntülerinden kullanıcı tarafından toplanan eğitim verileri ile eğitilen yapay sinir ağları ve ağaç yapılarının, hiç kullanılmayan görüntülerde iyi sonuçlar vermeleri mümkün müdür?”
- “Farklı uydu görüntülerinin SOM yöntemi sonuçları kullanılarak rastgele toplanan eğitim verileri ile eğitilen yapay sinir ağları ve ağaç yapılarının, hiç kullanılmayan görüntülerde iyi sonuçlar vermeleri mümkün müdür?”

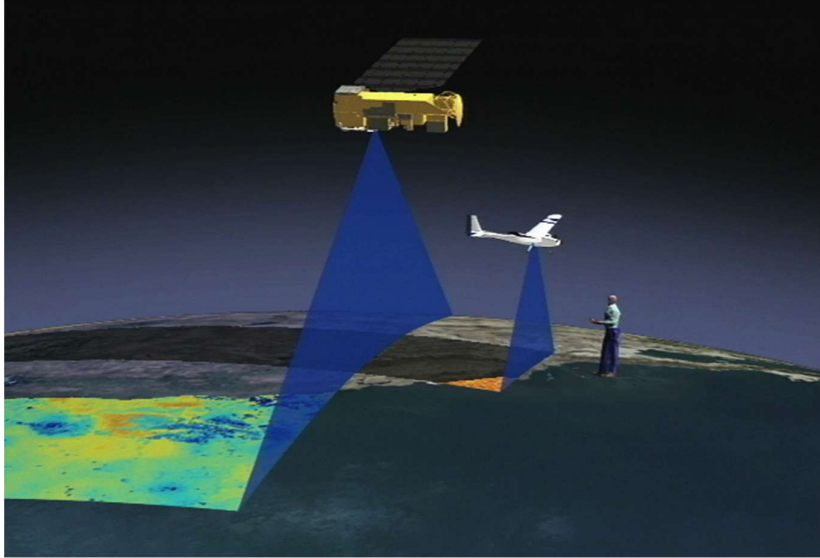
sorularının yanıtları araştırılmıştır.

### UZAKTAN ALGILAMA VE KIYI ÇİZGİSİ ÇIKARIMI

Bu bölümde uzaktan algılama ile kıyı çizgisi çıkarımında uzaktan algılamanın yeri ve öneminden bahsedilmiştir.

#### 2.1 Uzaktan Algılama

Uzaktan algılama, çeşitli platformlar (uydu, uçak, helikopter, insansız hava aracı vb.) kullanılarak yeryüzünde ya da uzayda bulunan cisimlerle direkt temas kurmadan bu cisimlerden yayılan-yansıyan enerjinin algılanması ve kaydedilmesidir. Uzaktan algılama sistemleri, özellikle uydulara yerleştirilen sistemler, yeryüzünün sık sık ve tutarlı bir şekilde görüntülerini elde ederler. Bu sistemler, yeryüzü sisteminin ve insan faaliyetlerinin dünyadaki etkilerini izlemek için çok değerlidir [19].



Şekil 2. 1 Uzaktan algılama platformlarına 2 örnek

Uzaktan algılama teknolojilerinin önemi uygulamalarından bazıları şunlardır [19]:

- Çevresel değerlendirme ve izleme (kentsel büyüme, tehlikeli atıklar)
- Küresel değişim algılama ve izleme (atmosferik ozon tüketimi, ormansızlaştırma, küresel ısınma)
- Tarım (mahsul durumu, verim tahmini, toprak erozyonu)
- Yenilemez kaynak araştırması (mineraller, petrol, doğalgaz)
- Yenilebilir doğal kaynaklar (sulak alanlar, topraklar, ormanlar, okyanuslar)
- Meteoroloji (atmosfer dinamiği, hava durumu tahmini)
- Haritalama (topoğrafya, arazi kullanımı, inşaat mühendisliği)
- Askeri gözetleme ve keşif (stratejik politika, taktik değerlendirme)
- Haber medyası (illüstrasyonlar, analizler)

## 2.2 Kıyı Çizgisi

Ülkemizde, yürürlükte olan 3621 ve bu kanunun bazı maddelerinde değişiklik yapılmasına dair 3830 Sayılı Kıyı Kanunu' na göre;

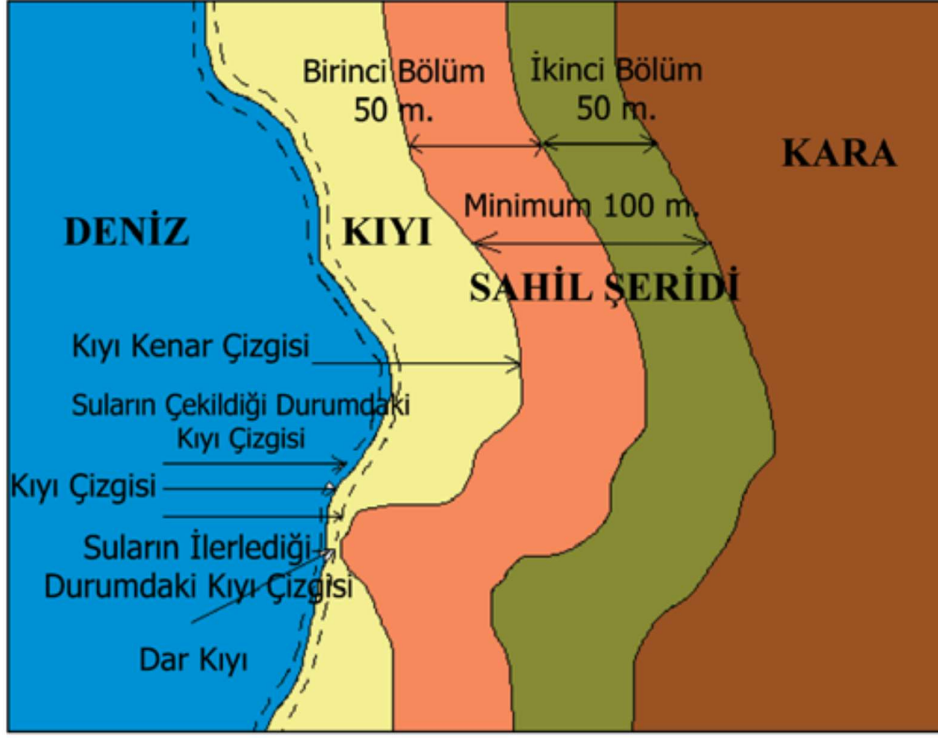
**Kıyı Çizgisi:** Deniz, doğal ve yapay göl ve akarsularda taşkın durumları dışında suyun kara parçasına değdiği noktaların birleşmesinden oluşan doğal çizgi,

**Kıyı Kenar Çizgisi:** Deniz, doğal-yapay göl ve akarsuların, kıyı özelliği gösteren kesimlerinde kıyı çizgisinden sonraki kara yönünde su hareketlerinin oluşturduğu alanların doğal sınırı,

**Kıyı:** Deniz, göl ve akarsuların kıyı çizgisi boyunca uzanan kıyı hareketlerinin düştüğü kumluk, çakıllık, taşlık ve bataklık alanlarının kara yönündeki doğal çizgisi arasında kalan alan,

**Sahil Şeridi:** Deniz, doğal ve yapay göllerin kıyı kenar çizgisinden itibaren kara yönünde yatay olarak en az 100 m genişliğindeki alan olarak tanımlanmaktadır.

**Kıyı Bölgesi:** Nehir deltaları, düz sahiller, kumsal ve kumullar, kayalıklar, bataklık ve lagünler vb. çok değişik yapılar içeren, sahil hattı çevresini kara kesiminde ve kıyı sularını da içerecek şekilde kucaklayan, karanın denizle birleştiği ortak yüzey olarak tanımlanabilmektedir [17].



Şekil 2. 2 Kıyı kanununa göre kıyı tanımlarını gösteren kroki [17]

Şekil 2. 2' de kıyı kanununa göre tanımlanan kıyılar yer almaktadır.

### 2.3 Kıyı Çizgisi Çıkarımında Uzaktan Algılamanın Yeri ve Önemi

Kıyılar, dünyada meydana gelen doğal ve yapay kökenli değişimlerin en çok yaşandığı alanlardır. Kıyı alanlarının deniz ve kara ortamlarının etkileşim sahasında olması meydana gelen doğal değişimlerin kıyı alanlarında etkili olmasına neden olmaktadır. Bunun yanında insanoğlunun tarih boyunca yerleşim, barınma, ekonomi ve ulaşım amaçlı olarak daima kıyı alanlarını kullanması, bu sahalarda yapay kökenli kıyı değişimlerinin de yaşanmasına neden olmuştur [20].

Kıyı çizgisi pozisyonunun bilinmesi, kıyısız problemlerle mücadelede, kıyının kara ve deniz tarafındaki kaynakların tanımlanmasında ve ölçülmesinde en temel gereksinimdir. Bu nedenle, kıyı çizgisinin veya deniz-kara sınırının belirlenmesi, kıyı çizgilerinin değişimlerinin tespiti, kıyı bölgesi yönetimi, havza sınırlarının çizilmesi ve taşkın koruma çalışmaları gibi birçok konu açısından önem taşımaktadır. Ancak, geleneksel yer ölçüm teknikleri ile kıyı çizgisinin veya deniz-kara sınırının belirlenmesi özellikle çok geniş alanlarda zaman alıcı, zor ve bazı durumlarda imkânsız bir iş olarak karşımıza çıkmaktadır. Uzaktan algılama teknikleri ve sağladığı verileri ise özellikle kıyı çizgisi değişimlerinin tahmininde, kıyısız alanların ve delta bölgelerin araştırılmasında ve izlenmesinde geniş alanlar için çok önemli ön bilgiler sağlayabilmektedir [21].

Son zamanlarda kıyı alanı yönetimi ve kıyı çizgisindeki ya da kıyı kullanımındaki değişimin izlenmesi amacıyla, uzaktan algılama ve coğrafi bilgi sistemleri yaygın bir şekilde kullanılmaktadır. Uzaktan algılama aynı zamanda kıyı değişiminde kullanılan en verimli yöntemlerden biridir [15].



### YÖNTEMLER

Kümelemenin ardındaki temel düşünce, nesnelerin kendi aralarındaki benzerlikleri göz önüne alınarak verilerin gruplandırılması işlemidir. Başka bir deyişle, birbirine benzeyen nesnelere ayırma işlemidir. Sınıflandırmanın kümelemeden farkı, önceden tanımlanmış sınıflar olmasıdır [22].

Kendini düzenleyen haritalar (SOM) yöntemi bir kümeleme yöntemi ve yapay sinir ağları (ANN) ile rastgele orman (RF) yöntemleri ise sınıflandırma yöntemleridir.

Bu bölümde ilk önce, sunulan çalışmada kullanılan yöntemler sırasıyla Kendini Düzenleyen Haritalar (SOM), Yapay Sinir Ağları (ANN) ve Rastgele Orman (RF) başlıkları altında anlatılmıştır. Daha sonra ise, bu yöntemlerden elde edilen kıyı çizgilerinin elle yapılan sayısallaştırma sonuçları ile karşılaştırılma işlemleri anlatılmıştır.

#### **3.1 Kendini Düzenleyen Haritalar (SOM)**

Kendini düzenleyen haritalar (self-organizing maps), kendini düzenleyen özellik haritaları, Kohonen ağları, Kohonen haritaları veya kısaca SOM olarak bilinen kendini düzenleyen haritalar (SOM) 1982 yılında Finlandiya Akademisi'nde profesör olan Tuevo Kohonen tarafından geliştirilmiştir [23].

SOM, kendini düzenleyen sinir ađlarını yani kontrolsüz yapay sinir ađlarını kullanarak veri boyutlarını azaltan bir veri görselleştirme tekniđidir. Veri görselleştirmedeki en büyük sorun insanların yüksek boyutlu verileri basitçe görselleştirememesidir. Bu nedenle, yüksek boyutlu verileri anlamalarına yardımcı olacak teknikler geliştirilmiştir. SOM yönteminin boyut indirgemedede izlediđi yol basit şekilde, benzer verileri bir araya getirerek bunları gruplaması şeklindedir. Dolayısıyla bu yöntemin boyut indirgeme ve benzerlikleri görselleştirme olarak kabaca 2 aşamadan oluştuđunu söylemek mümkündür [24].

SOM yöntemi, isminden de çok net bir şekilde anlaşıldıđı gibi, yöntemin kendi kendini organize etmesi, hiçbir denetlemeye ihtiyaç duymaması anlamına gelmektedir. SOM ađları, denetimsiz rekabetçi öğrenme ile kendi başlarına öğrenmeyi gerçekleştirmektedir. Yöntemin ismindeki haritalar kelimesi, ađrılık deđerlerini girdi verilerine uyacak şekilde eşleştirmeye çalıştıkları içindir. Bir SOM ađındaki düğümler (nodes), onlara sunulan girdiler gibi olmaya çalışmaktadır. Bu bağlamda, öğrenme işlemi böyle gerçekleştirilmektedir. Girdi verilerinin özellik verilerinin saklanması ilkesi, SOM yönteminin temel bir ilkesidir ve bu yöntemi çok deđerli kılan şeylerden biridir. Özellikle girdi verileri arasındaki topolojik ilişkiler, girdi verisi haritası oluşturulduktan sonra da korunmaktadır. Bu karmaşık verilerin sunumunda pragmatik bir deđere sahiptir [25].

SOM yönteminin temelde 2 bileşeni vardır: veri ve ađrılıklar. Veri bileşeni, çok boyutlu olan ve boyut indirgemesi yapılmak istenen girdi veridir. Ađrılıklar bileşeni ise, 2 bileşenden oluşur: ađrılık deđerleri ve bu deđerlerin girdi verideki konumları [24].

### **3.1.1 SOM' a Ait Öğrenme Algoritmaları**

SOM yönteminde temel olarak 2 farklı öğrenme algoritması kullanılmaktadır. Bunlar sıralı öğrenme ve toplu (batch) öğrenme algoritmalarıdır [26].

Sıralı öğrenmede, girdi verisindeki her bir veri ile en uygun eşleşen veri (Best Matching Unit – BMU) aranmaktadır ve hemen eşleştirilmektedir. Toplu öğrenme algoritmasında ise, bütün girdi veri için güncellemeler iterasyon sonuna saklanmaktadır [27].

### 3.1.1.1 Sıralı Öğrenme Algoritması

Sıralı öğrenme işleminin, [28]' e göre matematiksel modeli 2 adımda sunulmuştur.

**Adım 1:**  $b_i$ ,  $x_i$  ile eşleşen en iyi harita elemanı ve  $m_j$  ise  $x$  vektörünün  $j$  komşusuna ait vektör olmak üzere:

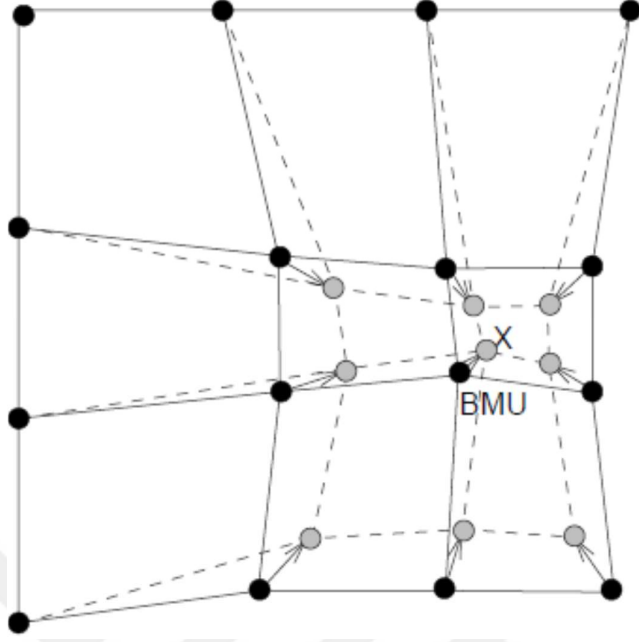
$$b_i = \arg \min_j \{ \|x_i - m_j(t)\| \} \quad (3.1)$$

denklemleri yardımıyla en yakın harita elemanı bulunur.  $t$  iterasyon sayısını,  $x_i$  eğitim kümesinden rastgele seçilen örnek veri vektörünü temsil etmektedir. En iyi harita elemanını bulmak için,  $x_i$  ile diğer tüm komşu vektörleri arasındaki mesafeler ölçülmektedir.

**Adım 2:**  $m_j$  vektörleri,  $b_j$  yönünde hareket ettirilerek güncellenir. Bu işlemin denklemi,

$$m_j(t + 1) = m_j(t) + \alpha(t)h_{b_{ij}}(t)[x_i - m_j(t)] \quad (3.2)$$

şeklinde dir. Burada  $t$  iterasyon sayısını,  $\alpha(t)$  öğrenme oranını ve  $h_{b_{ij}}(t)$  ise komşuluk fonksiyonu olup kazanan komşu birimin değerini temsil etmektedir. Bu komşu birim değeri kazanan komşuda en yüksek iken, mesafe arttıkça bu değer azalmaktadır.



Şekil 3. 1 X ile işaretlenen kazanan birimin ve komşularının konumlarını bu kazanan birime doğru güncellemesi.

Şekil 3. 1' de siyah nöronlar güncelleme öncesi, gri nöronlar güncelleme sonrası ve düz ile kesikli çizgilerde nöronlar arası topolojik ilişkileri temsil etmektedir [28].

### 3.1.1.2 Toplu (Batch) Öğrenme Algoritması

Toplu öğrenme algoritması iteratif bir algoritmadır ve tek bir veriye ait vektörü kullanmak yerine bütün veriyi herhangi bir ayarlama yapmadan önce ağa sunmaktadır [26].

Toplu öğrenme algoritmasının sıralı öğrenme algoritmasından bir farkı da ağırlık değerlerini hesaplarken öğrenme oranı ( $\alpha(t)$ ) kullanmamasıdır. Toplu öğrenme algoritması, oluşturulan harita ağında daha az güncelleme ve farklı iyileştirmeler yapılabildiğinden dolayı büyük veri setlerinde hesaplama işlemlerini daha hızlı gerçekleştirmek için kullanılmaktadır [27].

Toplu öğrenme algoritmasında da her iterasyonda (3.1) denklemi kullanılarak her bir  $x_i$  için en iyi harita elamanı ( $b_i$ ) bulunmaktadır. Daha sonra (3.3) denklemi kullanılarak her bir  $m_j$  vektörü,  $b_i$  yönünde hareket ettirilerek güncellenmektedir [28].

$$\mathbf{m}_j(t + 1) = \frac{\sum_{i=1}^N h_{bij}(t) \cdot x_i}{\sum_{i=1}^N h_{bij}(t)} \quad (3.3)$$

Bu denklemde  $N$ , toplam  $x_i$  vektörlerinin sayısını temsil etmektedir.

Orijinal SOM algoritması, sıralı öğrenme algoritması kullanılarak geliştirilmiştir. Fakat hesaplama işlemlerini hızlandırmaya yönelik araştırmalar sonucunda toplu öğrenme algoritmasının çok etkili bir şekilde bu hesaplama işlemlerini hızlandırdığı gözlemlenmiştir [29].

Sunulan tez çalışmasında Toplu Öğrenme algoritmasını kullanan SOM ağırları kullanılmıştır.

### 3.1.2 Toplu Öğrenme ile SOM Algoritması

SOM yönteminde nöronların kendilerini organize etme yolları, girdi verileri temsil etmek için birbirleriyle yarışmalarına dayanmaktadır. Nöronlar, her bir iterasyonda temsil hakkını kazanan nöron olmak için kendilerinin değiştirilmesine izin vermektedir. Ağırlıklarının kendilerini organize ederek benzerlikleri temsil eden bir haritaya dönüştürmesini sağlayan bu seçim ve öğrenme sürecidir [24].

Toplu öğrenme ile SOM algoritmasına ait işlem adımları aşağıda sıralanmıştır [26]:

Kendini düzenleyen haritalar,  $n \times m$  nörondan oluşan dikdörtgen bir grid ağından oluşmaktadır. Girdi vektör  $\mathbf{x}_j = (x_{j1}, x_{j2}, \dots, x_{jd})$  ( $j = 1, 2, \dots, N$ ) vektörü,  $d$  boyutlu her bir  $i$  nöronunun ağırlık değerlerinden oluşan  $\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{id})$  ( $i = 1, 2, \dots, n * m$ ) vektörü ise ağırlık vektörünü temsil etmektedir.

**ADIM 1:** İlk adımda ağırlık başlangıç ağırlık vektörleri rastgele belirlenir. Daha sonra her bir  $\mathbf{x}_j$  girdi vektörü ile bütün ağırlık vektörleri arasındaki mesafeler hesaplanır. Her bir  $\mathbf{x}_j$  girdi vektörüne en yakın nöron  $c_j$  (3.4) denklemi kullanılarak bulunur:

$$c_j = \arg \min_i \{ \|\mathbf{w}_i - \mathbf{x}_j\| \} \quad (3.4)$$

**ADIM 2:** Bu işlem iterasyon sayısı arttırılarak ( $t = t+1$ ,  $t =$  iterasyon sayısı) her bir girdi veriye en yakın nöron bulunur. Her bir girdi veriye ait en yakın nöron bulunduktan sonra girdi verilerin ağırlık değerleri (3.5) denklemi kullanılarak güncellenir:

$$w_i^{new} = \frac{\sum_{j=1}^N h_{c_j,i} x_j}{\sum_{j=1}^N h_{c_j,i}} \quad (3.5)$$

Bu denklemden  $h_{c_j,i}$  komşuluk fonksiyonudur ve

$$h_{c_j,i} = \exp\left(-\frac{\|r_i - r_{c_j}\|^2}{2\sigma^2(t)}\right) \quad (3.6)$$

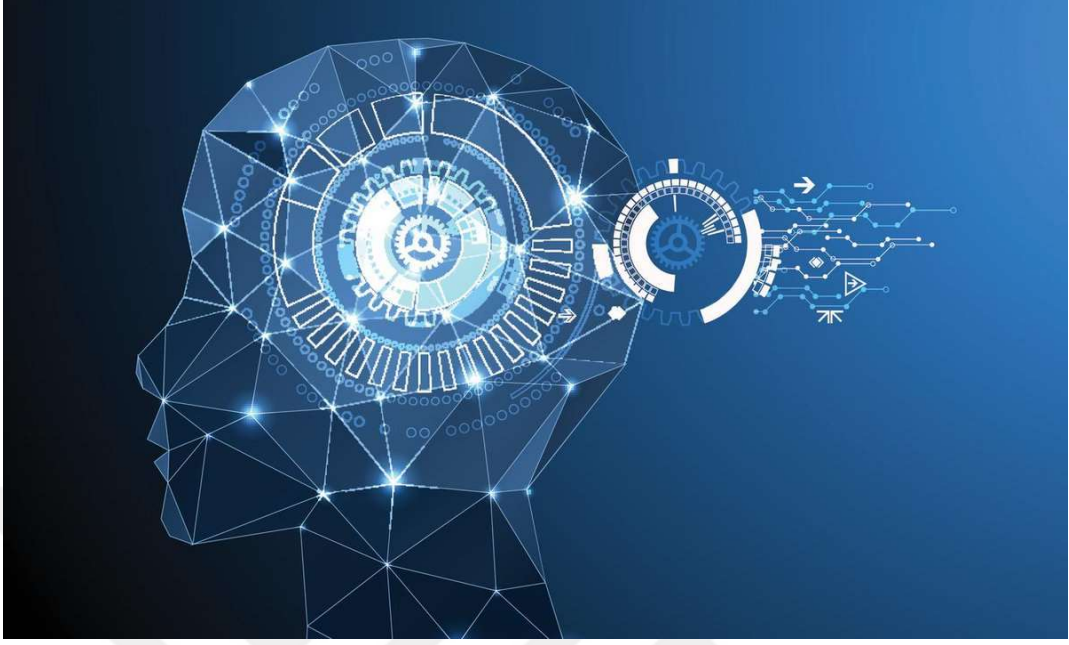
$\|r_i - r_{c_j}\|$  işlemi ile her bir  $j$  girdi verisine ait en yakın nöron  $c_j$  ile diğer her bir  $i$  nöronu arasındaki mesafe hesaplanır.  $\sigma(t) = \sigma_0(1 - t/t_{max})$  komşuluk yarıçap fonksiyonu ile öğrenme süresi azaltılır, burada  $\sigma_0$ , komşuluk yarıçapının ( $\sigma$ ) ilk değeri ve  $t_{max}$  maksimum iterasyon sayısıdır.

**ADIM 3:** Adım 1 ve Adım 2' deki işlemler iterasyon sayısı tamamlanıncaya kadar devam eder.

### 3.2 Yapay Sinir Ağları (Artificial Neural Networks – ANN)

Sinir ağları alanında yapılan çalışmalar son yıllarda giderek artan bir ilgi görmektedir. Warren McCulloch ve Walter Pitts 1943 yılında ilk yapay nöronların modelini sunduğundan beri, süregelen yıllarda yeni ve daha gelişmiş öneriler sunulmuştur. ANN, sinir sistemlerinin bilgi işleme yeteneklerini modellemeye yönelik bir girişimdir [30].

Diğer bir deyişle ANN, herhangi bir yardım almadan doğrudan insan beyninin öğrenme, hatırlama, üretme gibi özelliklerini kendinde barındırabilen ağ yapısındaki bir modeldir [31].

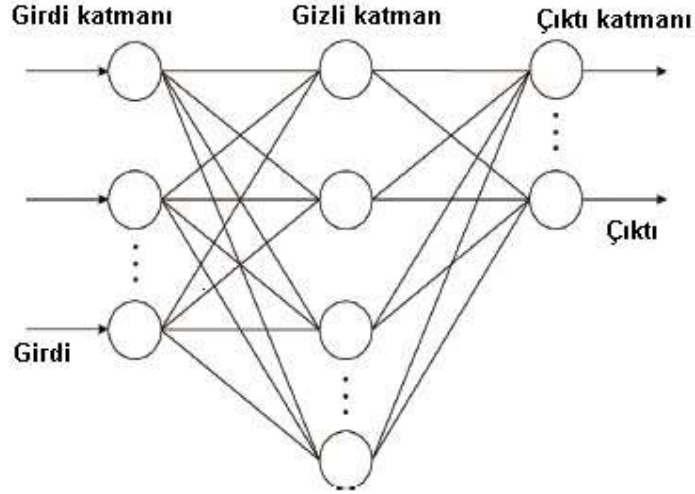


Şekil 3. 2 Yapay zekâ

ANN biyolojik sinir ađlarına benzer şekilde geliřtirilmiřtir. Biyolojik sinir hücresi sinapslar ile iletiřim kurup iřledikleri bilgileri de sahip oldukları aksonlar aracılıđıyla diđer hücrelerle paylařmaktadır. Benzer şekilde yapay olarak geliřtirilen sinir hücresi dıřarıdan gelen bilgileri bir toplama fonksiyonu ile toplar ve aktivasyon fonksiyonu ile bu bilgileri iřledikten sonra ađa ait bađlantıları kullanarak diđer hücrelere göndermektedir. Bir ANN yapısı 3 katmandan oluřmaktadır. Bunlar;

- Girdi katmanı
- Ara katmanlar
- Çıktı katmanı

Girdi katmanı, ađa gelen bilgilerin giriř yaptığı katmandır. Ađa giriř yapan bilgiler ara katmanlarda iřlenerek çıktı katmanına ulařtırılmaktadır [32].



Şekil 3. 3 ANN katmanları

Şekil 3. 3'te yapay sinir ağlarının yapısını oluşturan katmanlar gösterilmiştir.

### 3.2.1 Eğitim Yöntemleri

Yapay sinir ağlarını eğitmek için birçok eğitim yöntemi bulunmaktadır. Burada Gradient Descent, Conjugate Gradient ve Quasi-Newton yöntemlerine yer verilmiştir.

#### 3.2.1.1 Gradient Descent (Eğim Azaltımı) Yöntemleri

Temel gradient descent yöntemine göre nöronların ağırlıklarını performans fonksiyonunun gradyan değerinin azaldığı yönde güncelleyen en popüler eğitim yöntemleridir. Gradient descent eğitim yöntemleri geriye yayımlı, momentumlu ve esnek geriye yayımlı olmak üzere üç yöntemden oluşmaktadır [33].

#### 3.2.1.2 Conjugate Gradient (Birleşik Eğitim) Yöntemler

Temel gradient descent yöntemi ağırlıkları, performans fonksiyonunun gradyan değerinin en hızlı şekilde azaldığı yöne doğru ayarlamaktadır. Bu en hızlı şekilde yakınsama sağlayamamaktadır. Conjugate gradient yöntemleri, en çok azalma yönlerini bulmak yerine eşlenik yönler arayarak genellikle daha hızlı yakınsama sağlayan yöntemlerdir. Conjugate gradient yöntemleri diğer yöntemlere göre biraz daha fazla yer tutmaktadır. Bu nedenle, bu yöntemler çok sayıda ağırlığa sahip ağlar için iyidir.



Conjugate gradient eğitim yöntemleri, ölçekli, Fletcher-Reeves yenilemeli geri yayımlı ve Polak-Riebre yenilemeli geri yayımlı olmak üzere üç yöntemden oluşmaktadır [34].

Bu tez çalışmasında Ölçekli (Scaled) Conjugate Gradient Yöntemi kullanılmıştır.

#### **3.2.1.2.1 Scaled Conjugate Gradient (Ölçekli Birleşik Eğitim) Yöntemi**

Bu yöntem 1993 yılında Moller tarafından geliştirilen ölçekli conjugate gradient yöntemi, sayısal analiz problemlerinde çok kullanılmaktadır. Diğer birçok standart geriye yayılım yöntemlerinden farklı olarak, bu yöntemde kullanıcıdan herhangi bir parametre girişi beklenmemektedir ve hesaplama gücü diğer yöntemlere göre daha yüksektir [35].

Ölçekli conjugate gradient yönteminde, conjugate gradient yöntemlerinde olduğu gibi satır araması gerektirmemektedir. Her bir öğrenme iterasyonunda, satır araması yapmasını önleyen bir mekanizma kullanarak diğer yöntemlerden daha hızlı çalışmaktadır. Diğer conjugate gradient yöntemlerine göre daha fazla iterasyona ihtiyaç duymaktadır, fakat işlem sayısı her iterasyonda satır taraması yapılmadığı için önemli derecede azalmıştır [36].

Satır araması, bir hedef fonksiyonun yerel minimum değerini bulmaya yarayan iteratif bir yaklaşımdır [37].

#### **3.2.1.3 Quasi-Newton (Newton Benzeri) Yöntemleri**

Newton yöntemleri conjugate gradient yöntemlerine göre daha iyi ve hızlı optimizasyon sağlamaktadır. Newton yöntemlerinin temel basamağını, başarı ölçütündeki güncel ağırlık ve eğilim değerlerine ait Hessian matrisi (ikinci türevler) oluşturmaktadır. Newton yöntemleri conjugate gradient yöntemlerine göre daha hızlı yakınsamaktadır, fakat bu yöntemler daha karmaşıktır ve ileri beslemeli yapay sinir ağlarında Hessian matrisini hesaplamak daha fazla zaman gerektirmektedir. Newton yöntemlerinden ikinci türevin hesaplanmasını gerektirmeyenler Quasi – Newton ya da Secant yöntemleri olarak adlandırılmaktadır. Bu yöntemler algoritmalarının her bir iterasyonunda Hessian matrisini güncellemektedir. Quasi – Newton yöntemleri Broyde-Fletcher-Goldfarb-Shanno ve Levenberg-Marquardt geri yayılım yöntemlerinden oluşmaktadır [33].

Quasi – Newton yöntemleri ikinci türevleri kullanılmayan veya hesaplanması çok uzun işlemler gerektiren düzgün doğrusal olmayan fonksiyonların minimizasyonunda en etkili yöntemlerdir. Bu yöntemler, bir dizi arama yönü boyunca eğriliği tahmin ederek ikinci türev bilgilerini oluşturmaktadır. Bu yöntemlerden en başarılı yenileme yöntemi Broyde-Fletcher-Goldfarb-Shanno (BFGS) yöntemidir. Bu yöntemlerin çok çeşitli problemlerde başarısına rağmen, klasik Quasi – Newton yöntemleri çok büyük sayılarda iterasyon ve bazı problemlerde fonksiyon değerlendirmesi gerektirebileceği iyi bilinmektedir [38].

Bu tez çalışmasında Levenberg- Marquardt Geri Yayılım Yöntemi kullanılmıştır.

### **3.2.1.3.1 Levenberg- Marquardt Geri Yayılım Yöntemi**

Levenber – Marquardt geri yayılım yöntemi, doğrusal olmayan gerçek değerli fonksiyonların karelerinin toplamı olarak ifade edilebilen çok değişkenli bir fonksiyonun minimumunu bulmaktadır. Algoritmasının her iterasyonunda performans fonksiyon değerini daima azaltarak çalışan iteratif bir yöntemdir. Bu özelliği, Levenber – Marquardt geri yayılım yöntemini orta büyüklükteki ağlarda en hızlı eğitim algoritması yapmaktadır. Levenber – Marquardt geri yayılım yöntemi de Broyde-Fletcher-Goldfarb-Shanno yöntemi gibi fazla bellek ihtiyacı ve eğilimin ve Hessian matrisinin hesaplanmasının güçlüğü konusunda dezavantaja sahiptir [39].

## **3.2.2 Yapılarına Göre ANN Modelleri**

Farklı transfer fonksiyonları ve eğitim yaklaşımlarına bağlı olarak çeşitlilik gösteren yapay sinir ağlarının yapıları, içerdikleri nöronların birbirlerine bağlanma tiplerine göre ileri ve geri beslemeli olarak ikiye ayrılmaktadır [40].

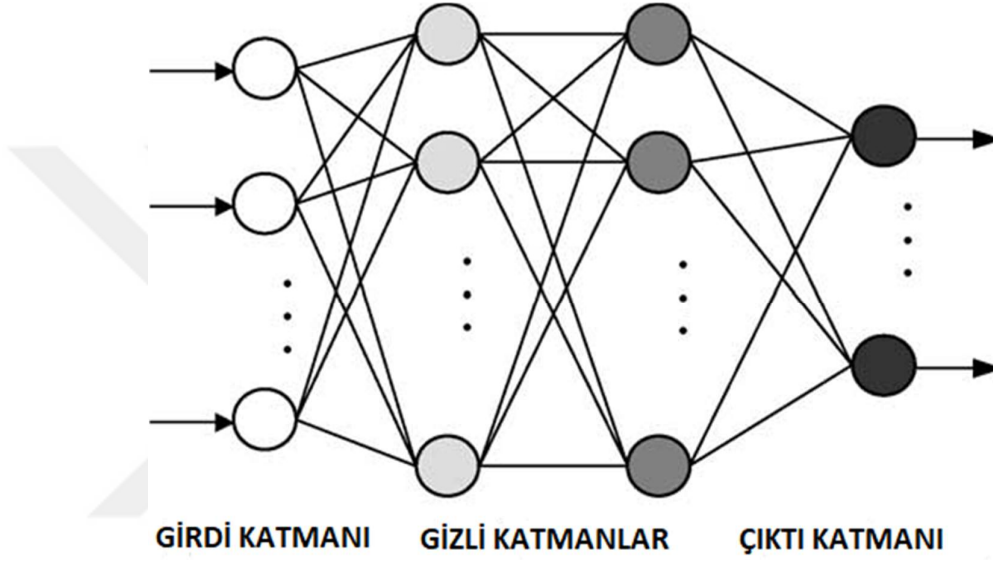
### **3.2.2.1 İleri Beslemeli Ağlar**

İleri beslemeli ANN, yapay sinir ağlarının en yaygın ve ilk geliştirilen türlerden biridir [41].

İleri beslemeli ağlarda nöronlar çıkışa doğru düzenli katmanlardan oluşmaktadır ve her birinden sadece kendinden sonraki katmanlara bağlantı vardır [40].

İleri beslemeli ANN, doğrusal çıktı katmanı ile birlikte bir veya daha fazla doğrusal olmayan gizli katmanlardan oluşmaktadır [42].

İleri beslemeli ANN modelinde, nöron ağırlıkları eğitimden sonra sabitlenir. Bu nedenle herhangi bir nöronun durumu, nöronun başlangıç ve geçmiş durumlarıyla değil, yalnızca girdi ve çıktı verileri ile belirlenmektedir. Yani herhangi bir dinamizm söz konusu değildir. Sonuç olarak, ileri beslemeli ağlar statik ağlar olarak da adlandırılabilir. İleri beslemeli sinir ağının avantajı, basit bir optimizasyon algoritması ile kolaylıkla oluşturulabilmesi ve günümüzde kullanılan en popüler sinir ağ mimarisi olmasıdır [43].



Şekil 3. 4 İleri beslemeli ANN [43]

Şekil 3. 4' te ileri beslemeli yapay sinir ağına ait girdi, gizli ve çıktı katmanları gösterilmiştir. Oklar ise ağın ileri beslemeli olduğunu belirtmektedir.

### 3.2.2.1.1 Çok Katmanlı Algılayıcı (Multilayer Perceptron) Ağlar

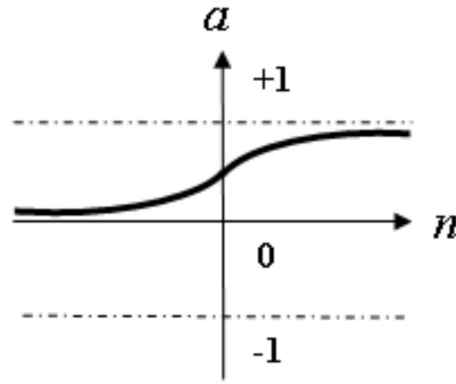
Çok katmanlı algılayıcı ağlar, ileri beslemeli yapay sinir ağlarının en sık kullanılan üyesidir. Bunun temel nedeni basit ve çok karmaşık fonksiyonel ilişkileri modellemesi yeteneğidir. Bu çok sayıda pratik uygulama vasıtasıyla kanıtlanmıştır. Çok katmanlı algılayıcı ağlar, algılayıcı (perceptron) adı verilen basit nöronlardan oluşan ağlardır. Bu algılayıcıların her biri girdi ağırlıklarına göre bir kombinasyon oluşturmaktadır ve bazı doğrusal olmayan aktivasyon fonksiyonları ile birden çok girdiden tek bir çıktı hesaplamaktadır [44].

Çok katmanlı algılayıcılara ait en sık kullanılan aktivasyon fonksiyonları [45],

- Linear aktivasyon fonksiyonu (PURELIN)
- Hyperbolic Tangent Sigmoid aktivasyon fonksiyonu (TANSIG)
- Logistic Sigmoid aktivasyon fonksiyonu (LOGSIG)
- Hard Limit aktivasyon fonksiyonu (HARDLIM)
- Radial Basis aktivasyon fonksiyonu (RDBAS)

Sunulan tez çalışmasında, Hyperbolic Tangent Sigmoid (TANSIG) ve Logistic Sigmoid (LOGSIG) aktivasyon fonksiyonları kullanılmıştır.

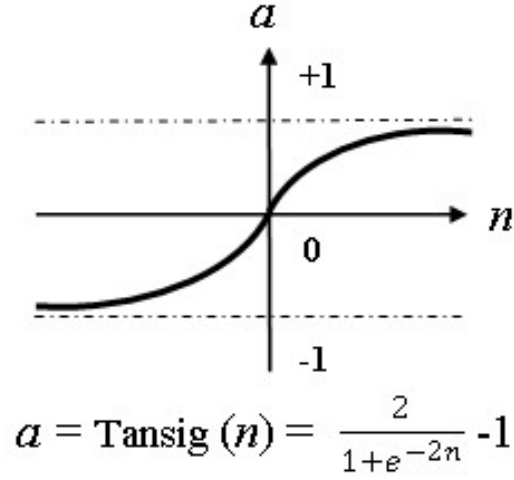
**Logistic Sigmoid aktivasyon fonksiyonu (LOGSIG):**  $[-\infty, +\infty]$  değer aralığında bulunan ve en sık kullanılan doğrusal olmayan aktivasyon fonksiyonudur.



$$a = \text{Logsig}(n) = \frac{1}{1 + e^{-n}}$$

Şekil 3. 5 Logistic Sigmoid aktivasyon fonksiyonu [46]

**Hyperbolic Tangent Sigmoid aktivasyon fonksiyonu (TANSIG):** Çok katmanlı algılayıcı ağların gizli katmanlarında kullanılan diğer bir sigmoid aktivasyon fonksiyonudur. Logistic Sigmoid fonksiyonu ile aynı denkleme sahiptir fakat değer aralığı  $[-\infty, +\infty]$  değil  $[-1, +1]$  şeklindedir.



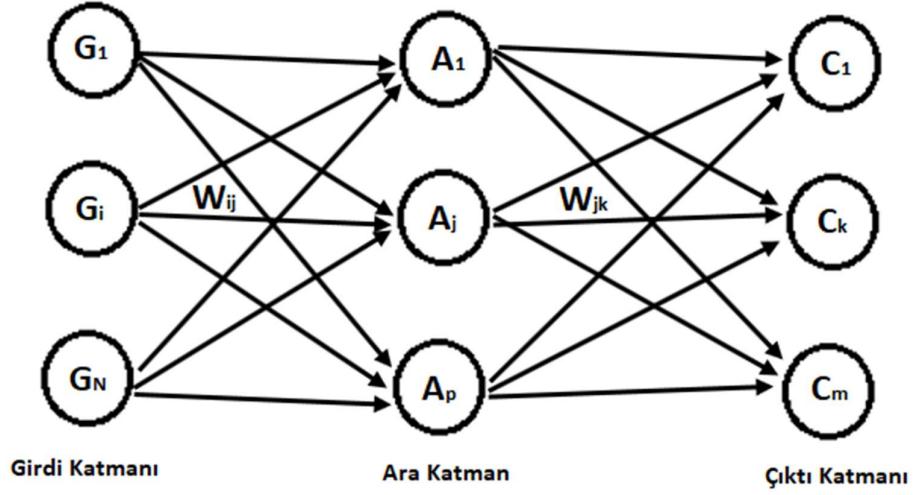
Şekil 3. 6 Hyperbolic Tangent Sigmoid aktivasyon fonksiyonu [46]

### 3.2.2.1.2 Geriye Yayılım Algoritması

Geri yayılım algoritması, en popüler algoritmalarından biridir. Bu algoritma; hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Bu yöntem, hataların geri yayılımı olarak da bilinmektedir [47].

Geriye yayılım algoritmasında, her bir girdi verisi gizli katmanlardaki nöronlarda işlenerek çıktı katmanında sonuç üretilmektedir. Elde edilen bu sonuçlar ile beklenen sonuçlar karşılaştırılarak hatalar bulunmaktadır. Çıktı katmanındaki hataların türevleri alınarak geriye doğru gizli katmanlardaki nöronlara aktarılmaktadır. Gizli katmanlardaki nöronlar gelen bu hataları azaltmak için kendi ağırlıklarını ayarlamaktadır. Bu ağırlıkları değiştirme denklemleri, ağda bulunan ortalama hatanın karesi en küçük olacak şekilde düzenlenmektedir [48].

Şekil 3. 7' de yer alan örnek ağ kullanılarak geri yayılım algoritması anlatılmıştır.



Şekil 3. 7 İleri beslemeli çok katmanlı ANN

$G_i$ : Girdi katmanına ait herhangi bir nöron

$g_i$ :  $G_i$  girdi katmanına gelen girdi verisi

$A_j$ : Ara katmana ait herhangi bir nöron

$a_j$ :  $A_j$  ara katmanına gelen veri

$C_k$ : Çıktı katmanına ait herhangi bir nöron

$c_k$ :  $C_k$  çıktı katmanına gelen çıktı verisi

$W_{ij}$ : Girdi katmanı ile ara katmandaki nöronlar arasındaki ağırlık değeri

$W_{jk}$ : Ara katman ile çıktı katmanındaki nöronlar arasındaki ağırlık değeri

$h_k$ : Hedef çıktı verisi

olmak üzere, geri yayılım algoritması işlem basamakları aşağıda sıralanmıştır [49]:

- **Basamak 1:** Ağırlıklara ( $W_{ij}$ ,  $W_{jk}$ ) rastgele sayılar atanır.
- **Basamak 2:** Eğitim için kullanılan girdi ve çıktı verilerinin tamamı için **Basamak 2** ile **Basamak 9** arasındaki işlemler gerçekleştirilir.
- **Basamak 3:** Her bir girdi verisi için **Basamak 3** ile **Basamak 8** arasındaki işlemler gerçekleştirilir.
- **Basamak 4:** Her bir  $G_i$  ( $G_1, G_2, \dots, G_N$ ) girdi katman nöronu kendisine gelen  $g_i$  girdi verisini alıp ara katmanlarda yer alan nöronlara aktarır.
- **Basamak 5:** Her bir  $A_j$  ( $A_1, A_2, \dots, A_p$ ) ara katman nöronu ağırlıklandırılmış girdi verilerini toplar ve çıktı nöronlara verilecek veriyi hesaplar. Hesaplanan bu çıktı verileri çıktı katmanındaki nöronlara gönderir.

$$a = W_{ij} + \sum g_i W_{ij} \quad (3.7)$$

$$a_j = f(a) \quad (3.8)$$

- **Basamak 6:** Her bir  $C_k$  ( $C_1, C_2, \dots, C_m$ ) çıktı katman nöronu ara katmandan gelen ağırlıklandırılmış verileri toplar ve çıktı verisini hesaplar.

$$c = W_{jk} + \sum a_i W_{jk} \quad (3.9)$$

$$c_k = f(c) \quad (3.10)$$

- **Basamak 7:** Her bir  $C_k$  ( $C_1, C_2, \dots, C_m$ ) çıktı nöronu, her bir girdi verisine karşılık gelen hedef çıktı verisini alır ve hata terimini hesaplar.

$$\delta_k = (h_k - c_k) f'(c) \quad (3.11)$$

Sonra  $W_{jk}$  ağırlık değerini güncellemede kullanılacak ağırlık düzeltme terimini hesaplar.

$$\Delta W_{jk} = \alpha \delta_k a_j \quad (3.12)$$

- **Basamak 8:** Ara katmana ait her bir  $A_j$  ( $A_1, A_2, \dots, A_p$ ) nöron, çıktı katmanından gelen hata değerlerini toplar.

$$\delta = \sum \delta_k W_{jk} \quad (3.13)$$

Sonra hata terimini hesaplar.

$$\delta_j = \delta f'(a) \quad (3.14)$$

Sonra  $W_{ij}$  ağırlık değerini güncellemede kullanılacak ağırlık düzeltme terimini hesaplar.

$$\Delta W_{ij} = \alpha \delta_j g_i \quad (3.15)$$

- **Basamak 9:** Her bir  $C_k$  ( $C_1, C_2, \dots, C_m$ ) çıktı nöronu, ağırlık değerlerini günceller.

$$W_{jk} (\text{yeni}) = W_{jk} (\text{eski}) + \Delta W_{jk} \quad (3.16)$$

$$W_{ij} (\text{yeni}) = W_{ij} (\text{eski}) + \Delta W_{ij} \quad (3.17)$$

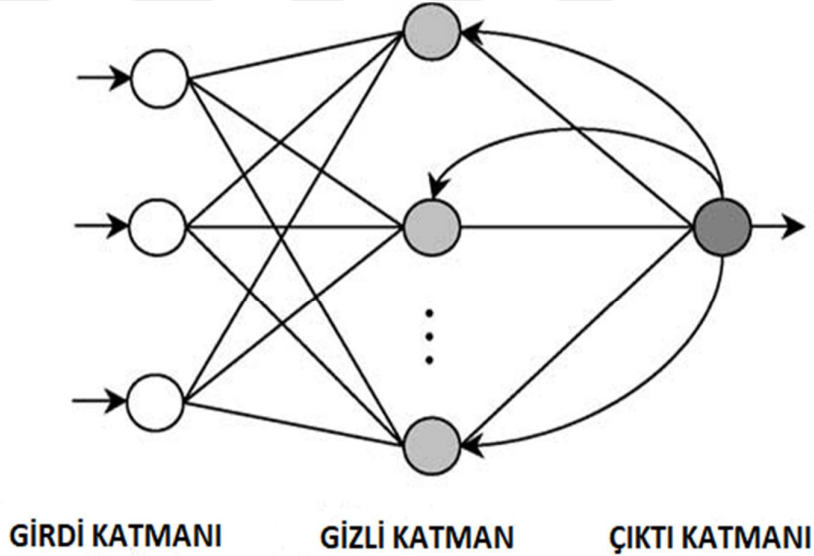
- **Basamak 10:** Bitiş koşulunu kontrol et.

Geri yayılma algoritmasından yukarıdaki 10 basamaktan oluşan döngüyü, her tamamladığında oluşan toplam ağ hatasının bir önceki döngüdekine göre daha küçük hesaplaması beklenmektedir. Böylece toplam hata minimize edilene kadar veya kabul edilebilir bir değere ulaştığında döngü sonlandırılacaktır [50].



### 3.2.2.2 Geri Beslemeli Ağlar

Çıktı katmandan girdi katmanına veya gizli katmandan girdi katmanına gibi birkaç farklı geri beslemeli ağ mimarileri mevcuttur. Başka bir deyişle, geri beslemeli ağlar, en az bir geriye dönük bağlantıya sahip olmaları ile ileri beslemeli ağlardan ayrılmaktadır. Nöronların bir veya birden fazla bağlantılarının durumu zamanla değişen bir yapıya sahip olduğundan, geri beslemeli ağlara aynı zamanda dinamik yapay sinir ağları da denmektedir. Geri beslemeli bir bağlantının var oluşu, ağın öğrenme kabiliyetini ve performansı üzerinde derin bir etkiye sahiptir. Geri yayımlı ağlara ait nöronların ağırlıkları ayarlanabilir olduğu için nöronların durumu ağa ilk girdikleri durumla beraber önceki durumlarına da bağlıdır. Geri beslemeli sinir ağının avantajı, ağın girdi verisinin boyutunu ve dolayısıyla eğitim süresini azaltabilmesidir [43].



Şekil 3. 8 Geri beslemeli ANN [43]

Şekil 3. 8' de geri beslemeli yapay sinir ağına ait girdi, gizli ve çıktı katmanları gösterilmiştir. Oklar ise geri beslemeli olduğunu belirtmektedir.

### **3.2.3 Öğrenme Türlerine Göre ANN Modelleri**

Günümüze kadar çeşitli öğrenme algoritmaları geliştirilmiştir. Bunlar temel olarak denetimli öğrenme, denetimsiz öğrenme ve takviyeli öğrenme olarak 3 ana gruba ayrılır [51].

#### **3.2.3.1 Denetimli Öğrenme**

Genellikle sınıflandırma ve regresyon problemlerinde kullanılan denetimli öğrenme, girdi ve çıktılarının bilindiği eğitim verisini içeren öğrenmedir [41].

Diğer bir deyişle, sinir ağına hem girdi hem de çıktı değerleri verilmektedir ve ağıın ürettiği çıktı değerleri ile istenen çıktı değerleri arasındaki fark sıfır veya ona yakın bir değere gelinceye kadar ağırlıklar değiştirilmektedir [51].

#### **3.2.3.2 Denetimsiz Öğrenme**

Genellikle kümeleme problemlerinde kullanılan denetimsiz öğrenme, girdilerin bilindiği fakat çıktıların bilinmediği eğitim verisini içeren öğrenmedir. Denetimsiz öğrenmede yapay sinir ağı, girdileri doğru bir şekilde birleştirerek uygun çıktıyı oluşturacak şekilde eğitilmektedir [41].

Diğer bir deyişle, ağa sadece girdi verisi verilmektedir. Girdi verisi uygun bir çıktı verisi üretinceye kadar nöronlar arası ağların ağırlıkları değiştirilmektedir [51].

#### **3.2.3.3 Takviyeli Öğrenme**

Takviyeli öğrenme, bulunduğu ortamı algılayan ve kendi başına kararlar alabilen bir sistemin, hedefine ulaşabilmesinde doğru kararlar almasını öğrenerek ağıın eğitildiği bir öğrenme çeşididir. Bu yöntem robotik, oyun programlama, hastalık teşhisi ve fabrika otomasyonu gibi alanlarda sıklıkla kullanılmaktadır. Takviyeli öğrenmede çıktı veri direkt ağıa sunulmamaktadır. Onun yerine ağı girdi verileri ile eğitilirken verdiği kararların doğru olduğu durumlarda ağı ödüllendirilmektedir ve yanlış olduğu durumlarda ağı cezalandırılmaktadır [52].

### 3.3 Rastgele Orman (RF)

Karar ağaçları (KA) son yıllarda literatürde yaygın kullanımı olan bir sınıflandırma ve örüntü tanımlama algoritmasıdır. Bu yöntemin yaygın olarak kullanımının en önemli nedeni ağaç yapılarının oluşturulmasında kullanılan kuralların anlaşılabilir ve sade olmasıdır. KA sınıflandırma işleminin gerçekleştirilmesinde çok aşamalı veya ardışık bir yaklaşım kullanılmaktadır. Yöntem uydu görüntülerinin sınıflandırılmasında olduğu gibi çok karışık bir sınıflandırma problemini aşamalı bir hale getirerek basit bir karar verme işlemi gerçekleştirmektedir [53].

Girdi veri farklı özelliğe sahip niteliklerden oluştuğundan dolayı karar ağaçlarında dallanmanın hangi kritere göre yapılacağı oldukça önemlidir. Örneğin renk, şekil, tat gibi girdi verisindeki verilerin özelliklerini tanımlayan nitelikler olabilir. Nitelik seçim kriterleriyle ağaç yapısındaki dallanmanın hangi niteliğe göre yapılacağı belirlenmektedir. En yaygın kriterler entropi, bilgi kazanımı ve GINI indeksidir. ID3 ve C4.5 algoritmaları entropi ve bilgi kazanımına dayalı bölünmelerde kullanılırken, Twoing ve GINI algoritmaları Sınıflandırma ve Regresyon Ağaçları (CART) yöntemine dayalı bölünmelerde kullanılmaktadır [54].

#### 3.3.1 Sınıflandırma ve Regresyon Ağaçları (CART)

CART, sınıflandırma ve regresyon ağacının kısaltmasıdır. CART algoritması, Morgan ve Sonquist' in AID (Automatic Interaction Detection) adlı karar ağacı algoritmasının devamı niteliğinde Breiman ve arkadaşları tarafından 1984 yılında önerilmiştir. Hem sayısal hem de nominal veri türlerini, girdi ve kestirimsel değişken olarak kabul edebilen CART algoritması, sınıflandırma ve regresyon problemlerinde bir çözüm olarak kullanabilmektedir. CART karar ağacı, ikili olarak özyinelemeli biçimde bölünen bir yapıya sahiptir. Dallanma kriteri olarak GINI indeksinden yararlanan CART ağacı, kuruluş aşamasında herhangi bir durma kuralı olmaksızın sürekli olarak büyümektedir. Artık yeni bir bölünmenin gerçekleşmeyeceği durumda bu sefer uçtan köke doğru budama işlemi başlatılmaktadır. Olası en başarılı karar ağacı her budama işlemi sonrası bağımsızca seçilmiş bir test verisi ile değerlendirme yapılarak tespit edilmeye çalışılmaktadır. Her bir düğümün her aşamada ikiye ayrıldığı CART algoritmasında, her bir bölünme noktasının

belirlenmesinde GINI, Twoing gibi en iyi bölmeyi seçmek için geliştirilen söz konusu homojenlik ölçütlerinden GINI indeksi kullanılmaktadır [55].

### 3.3.1.1 GINI İndeksi

GINI indeksi İtalyan ekonomist Corrado Gini tarafından ilk başlarda gelir eşitsizliğini ölçmek için kullanılmış olup, her bir girdi verinin homojenliğinin ölçüldüğü bir yöntemdir. Düşümlerdeki özniteliklerin tek bir sınıfa girmesi durumunda GINI indeksi minimuma ulaşır yani sıfır olur. Karar ağaçlarının bölünme kriteri, en düşük GINI indeksine göre girdi verilerini ayırmaktır [56].

GINI indeksi, ana düğüm sol ve sağ olarak 2 alt düğüme ayrıldıktan sonra hem sağ hem de sol düğümdeki nitelikler için ayrı ayrı hesaplanmalıdır. Bunun için,

$$G_{sağ}(T) = 1 - \sum_i p^2(x_i) \quad (3.18)$$

$$G_{sol}(T) = 1 - \sum_i p^2(x_i) \quad (3.19)$$

denklemleri kullanılır. Bu denklemlerde  $p(x_i)$   $T$  düğümündeki  $x_i$  niteliğinin olma ihtimalini temsil etmektedir. Sağ ve sol GINI değerleri hesaplandıktan sonra ana düğümün GINI indeksini hesaplamak için

$$G(x_i) = \frac{n_{sağ}}{n_{x_i}} \cdot G_{sağ} + \frac{n_{sol}}{n_{x_i}} \cdot G_{sol} \quad (3.20)$$

denklemini kullanılmaktadır. Bu denklemde  $n_{sağ}$  ve  $n_{sol}$  sağ ve sol düğümlerdeki toplam nitelik sayısını,  $n_{x_i}$  ise  $x_i$  ana düğümdeki nitelik sayısı temsil etmektedir [54].

### 3.3.2 Rastgele Orman (RF) Algoritması

RF (Random Forest) yönteminde birçok sınıflandırma ağacı bulunmaktadır. Girdi verinin her biri bu sınıflandırma ağaçlarının hepsinden geçirilmektedir. Sınıflandırma ağaçları ile girdi verinin her bir elemanı sınıflandırılmaktadır. Her bir girdi veri bütün sınıflandırma ağaçlarına girip oylandıktan sonra ağaç yapılarından çıkan sonuçlardan en fazla oyu alan sınıfa atamalar yapılmaktadır. Bu sınıflandırma ağaçlarının her biri aşağıdaki adımları takip etmektedir [57]:

- N sayıda eğitim verisi için orijinal veriden rastgele N sayıda veri toplanır.
- Eğer M tane girdi veri var ise, her bir düğüm için  $m < M$  olacak şekilde bir m sayısı belirlenir. m sayısı kadar veri M girdi verisinden rastgele seçilir ve bu düğümlerin en iyi şekilde ayrılması için kullanılır. Ağaç yapılarının oluşturulmasında bu m değerleri sabit kalır.
- Her ağaç mümkün olan en büyük ölçüde gelişir. Budama yapılmaz.

Ağaç yapıları gelişirken oluşabilecek orman hata oranı 2 şeye bağlıdır [57]:

- Ormandaki 2 ağaç arası korelasyon ve
- Ormandaki her bir ağacın dayanıklılığı

Oluşturulan ormandaki 2 ağaç arası korelasyonun artması orman hata oranının artması demektir. Her bir ağacın bireysel dayanıklılığının arttırılması orman hata oranını düşürecektir.

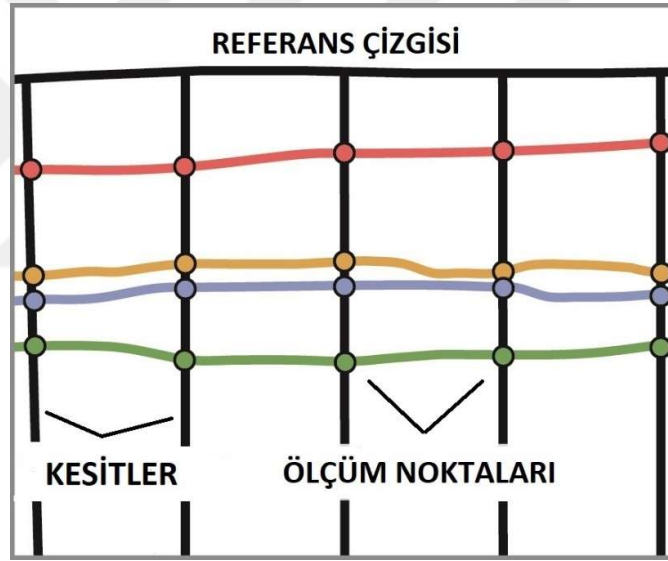
m sayısının azaltılması korelasyon ve dayanıklılığın azalmasını, arttırılması ise artmasını sağlamaktadır. En uygun m aralığı genellikle oldukça geniştir. Oob (out-of-back) hata oranı kullanılarak aralıktaki m değeri hızlıca bulunabilmektedir [57].

RF yönteminde, ağaç yapılarını oluşturmak için kullanılan verilerin  $2/3'$  ü eğitim,  $1/3'$  ü de test amaçlı kullanılmaktadır. Bu oransal dağılım kullanılan veri seti içerisine rastgele gerçekleştirilmektedir [4].

### 3.4 Sayısal Kıyı Çizgisi Analizi Sistemi (Digital Shoreline Analysis System – DSAS)

Uygulama bölümünde elde edilen kıyı çizgilerinin elle yapılan sayısallaştırma sonuçlarıyla karşılaştırılması işlemleri Sayısal Kıyı Çizgisi Analizi Sistemi (DSAS) kullanılarak yapılmıştır.

Sayısal kıyı çizgisi analizi sistemi (DSAS), 2 farklı kıyı çizgisi arasındaki değişim oranı istatistiklerinin hesaplanmasını sağlayan, Amerika Birleşik Devletleri Jeoloji Araştırma Kurumu (U. S. Geological Survey – USGS) tarafından geliştirilmiş olan ve Esri ArcGIS v.10 yazılımına eklenebilen bir fonksiyon paketidir. DSAS, kıyı boyunca kullanıcı tarafından belirlenen bir aralıkta referans çizgisine dikey olarak geçirilen kesitler üretir. DSAS kesitinde referans çizgisi ile her kıyı şeridi kesişim noktası arasındaki mesafeler hesaplanır [58].



Şekil 3. 9 DSAS bileşenleri

Şekil 3. 9' da DSAS bileşenleri olan referans çizgisi, kesitler ve ölçüm noktaları yer almaktadır. Kıyı çizgilerine ait ortalama hataları hesaplamak için ilk önce, referans çizgisine dik olan kesitler üzerindeki ölçüm noktaları arasındaki mesafeler hesaplanır. Daha sonra, bu hesaplanan mesafeler içerisinde uyumsuz ölçüler (kaba hata) ayıklanır ve kalan mesafe değerlerinin ortalaması alınır.

Kaba hata ayıklama işlemi için aşağıdaki işlem adımları izlenmiştir [59].

**Adım 1:** Hesaplanan deęişim deęerlerinin ortalamaları hesaplanır.  $n$  sayıdaki veri setinin  $l_i$  ölçülerinin ortalaması ( $\mu$ ),

$$\mu =_{n \rightarrow \infty} \frac{\sum_{i=1}^n l_i}{n} \quad (3.21)$$

denklemleri ile hesaplanır.

Ortalama deęerleri hesaplandıktan sonra her bir deęişim bu ortalamalardan çıkartılarak her bir ölçünün hatası ( $\varepsilon_i$ ) hesaplanır.

$$\varepsilon_i = \mu - l_i \quad (3.22)$$

**Adım 2:** Hesaplanan deęişim deęerlerinin standart sapmaları hesaplanır.

Standart sapma ( $\sigma$ ), (4.7)'deki denklem ile bulunur.

$$\sigma =_{n \rightarrow \infty} \sqrt{\frac{\sum_{i=1}^n \varepsilon_i^2}{n}} \quad (3.23)$$

Hesaplanan ölçü hatalarının  $-\sigma$  ile  $+\sigma$  arasında olma olasılığı;

$$P(-\sigma < \varepsilon_i < +\sigma) = 0,683 \quad (3.24)$$

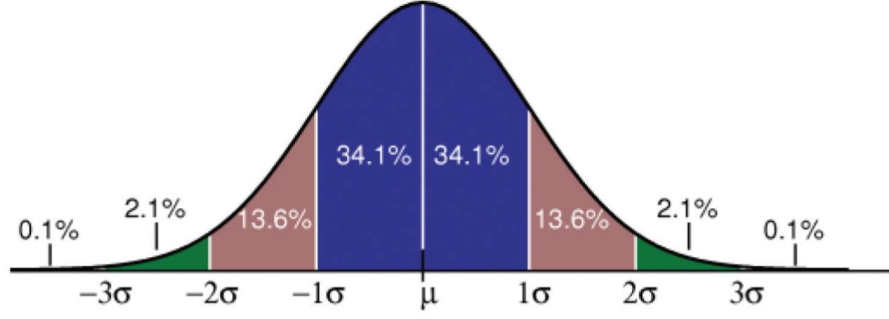
$-2\sigma$  ile  $+2\sigma$  arasında olma olasılığı;

$$P(-2\sigma < \varepsilon_i < +2\sigma) = 0,955 \quad (3.25)$$

$-3\sigma$  ile  $+3\sigma$  arasında olma olasılığı;

$$P(-3\sigma < \varepsilon_i < +3\sigma) = 0,997 \quad (3.26)$$

şeklindedir. Burada ölçü hatalarının mutlak deęerce standart sapmalarının 3 katından küçük olma olasılığının en yüksek olduęu görülmektedir. [59].



Şekil 3. 10 Hata olasılık fonksiyonu ve standart sapma [59]

(3.26) denklemindeki olasılık değeri en yüksek olduğu için, her bir kıyı çizgisinin ölçü hataları içerisinde standart sapmalarının 3 katından büyük olanlar kaba hata olarak kabul edilmiş ve elimine edilmiştir. Kaba hata eliminasyonu gerçekleştirildikten sonra geriye kalan mesafe değerlerinin ortalamaları alınarak kıyı çizgilerinin ortalama hataları yani DSAS sonuçları bulunmuştur.



### UYGULAMA

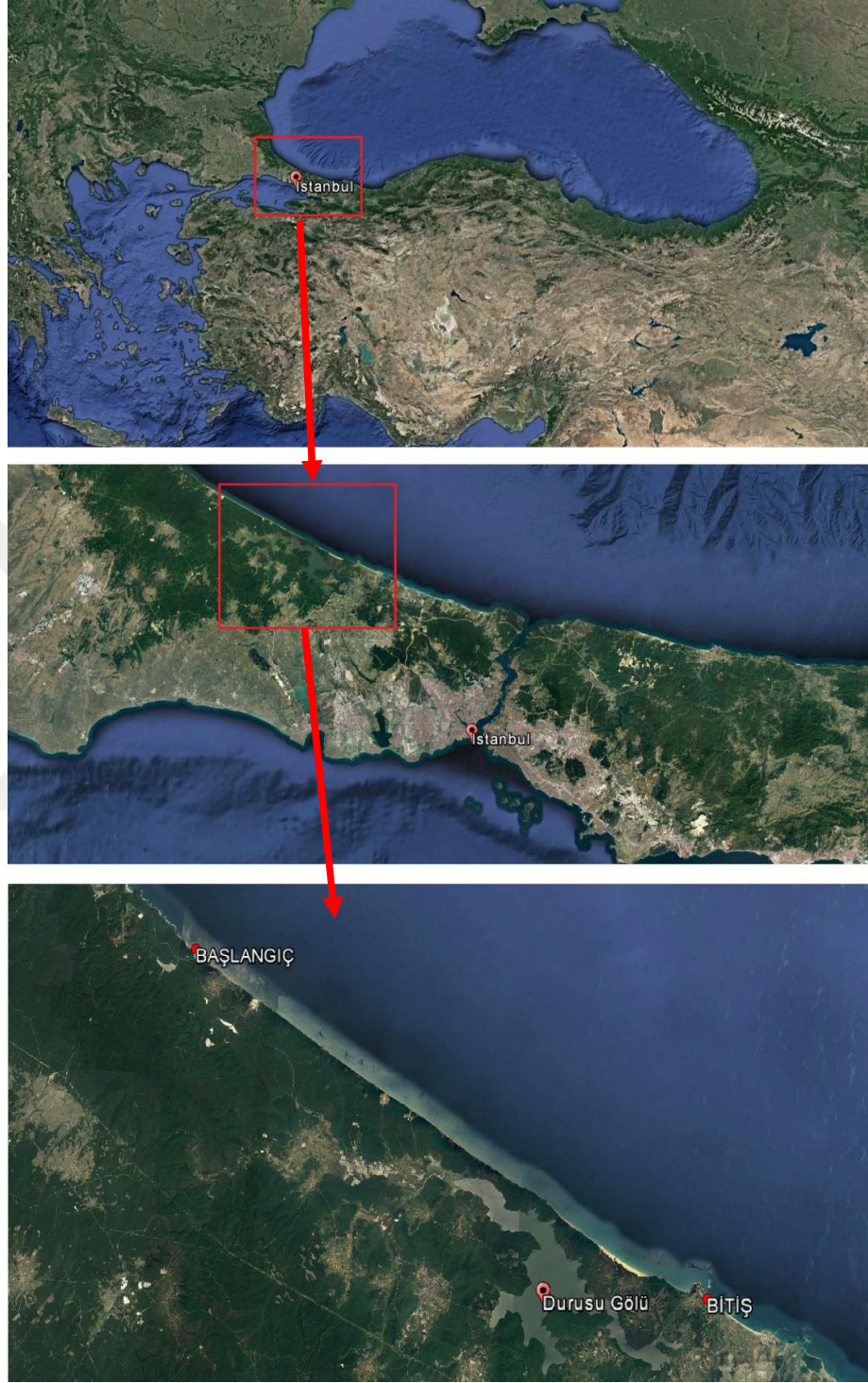
Bu bölümde, sunulan tez kapsamında gerçekleştirilen uygulamaya ait bilgilere yer verilmiştir. Birinci bölümde çalışma bölgeleri, ikinci bölümde çalışmanın gerçekleştirildiği algoritmalar ve donanım, üçüncü bölümde çalışmada kullanılan veriler ve dördüncü bölümde ise uygulamanın işlem adımları anlatılmıştır.

#### 4.1 Çalışma Bölgeleri

Yapılan uygulamada çalışma bölgelerinden biri olan Terkos bölgesi, İstanbul ilinin Çatalca ilçesi sınırları içerisinde yer almaktadır. İçerisinde İstanbul' un en büyük gölü olan Terkos (Durusu) Gölü'nün de bulunduğu bu bölge, İstanbul ilinin en önemli su havzalarından biridir.

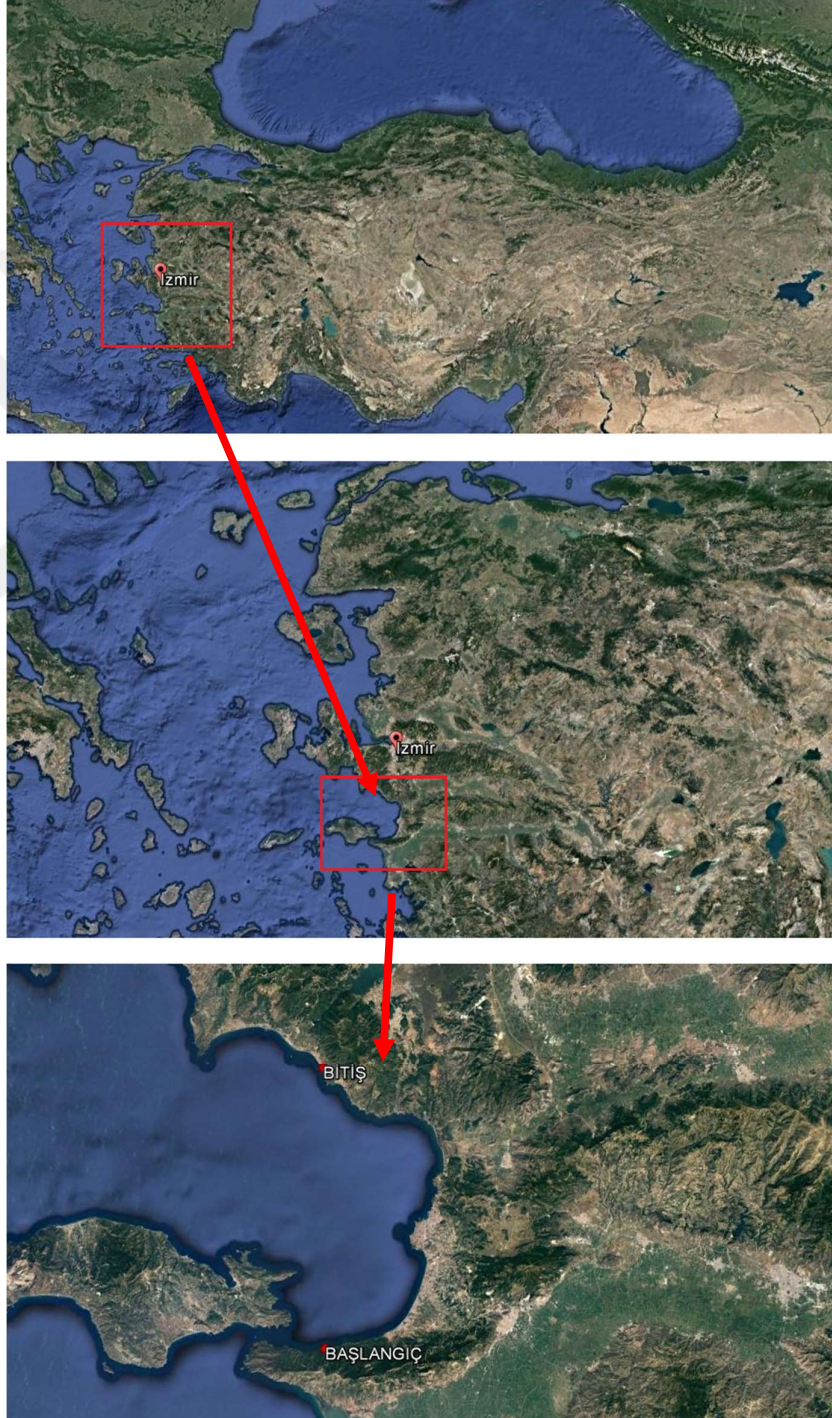
Uydu görüntüleri ile Terkos-Karadeniz arası kıyı değişiminin aşırı olduğu tespit edildikten sonra dolgu çalışmaları başlatılmıştır. İstanbul ilinin su ihtiyacının %35' ini Terkos Gölü sağlamaktadır. Geçmiş yıllarda kaçak kum çekimi nedeniyle, Karadeniz ile birleşme tehlikesi göz önünde bulundurularak kullanıma yasak bölge ilan edilmiştir [60].

Bu nedenle Terkos bölgesi, birçok çalışmada uygulama alanı olarak kullanılmıştır ve sunulan tez çalışmasında da bu bölge seçilerek Karadeniz kıyısının en hızlı ve otomatik bir şekilde tespit edilmesi amaçlanmıştır.

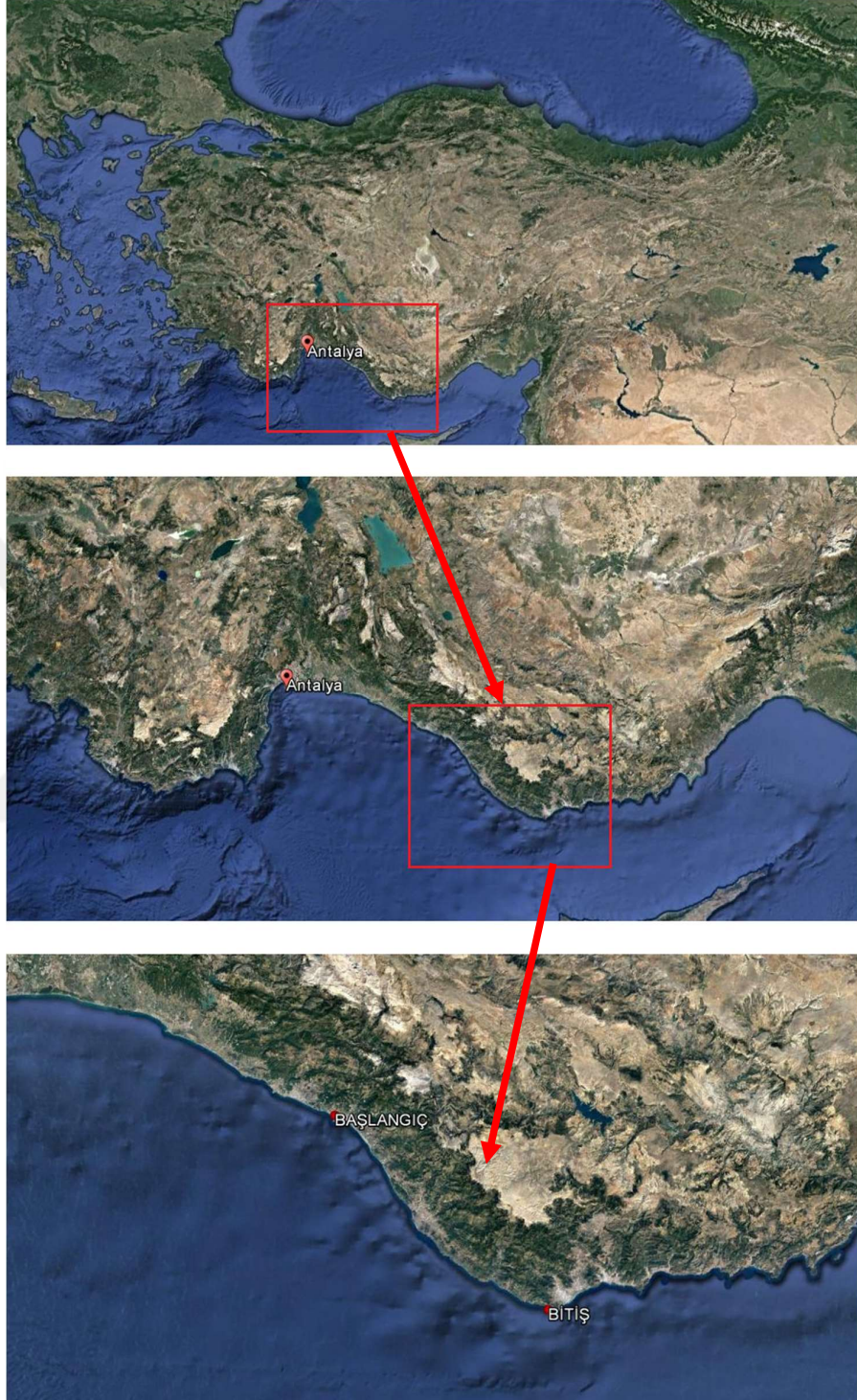


Şekil 4. 1 Terkos çalışma bölgesinin Google Earth üzerindeki görüntüsü

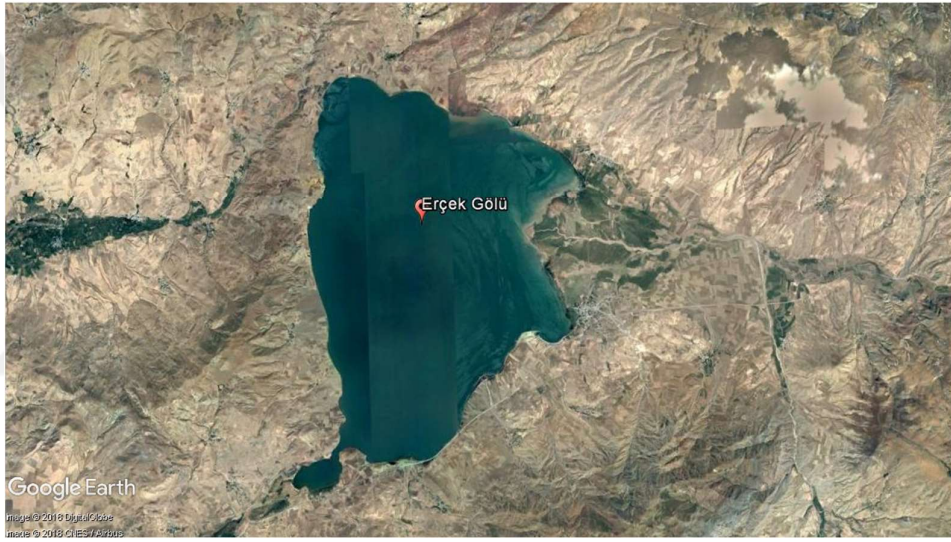
Terkos bölgesi dışında, İzmir ve Antalya bölgeleri ile Erçek Gölü çalışma alanı olarak seçilmiştir. Terkos, İzmir ve Antalya bölgelerinin, elle yapılan sayısallaştırmaya göre sırasıyla 42,138 km, 45,760 km 40,862 km' lik kısımları kullanılmıştır. Erçek Gölü' ne ait elle yapılan sayısallaştırma ise, 55,051 km uzunluğuna sahiptir.



Şekil 4. 2 İzmir çalışma bölgesinin Google Earth üzerindeki görüntüsü



Şekil 4. 3 Antalya çalışma bölgesinin Google Earth üzerindeki görüntüsü



Şekil 4. 4 Erçek Gölü' nün Google Earth üzerindeki görüntüsü

## 4.2 Kullanılan Programlar ve Donanım

Sunulan tez kapsamında uygulamaları gerçekleştirmek için,

- ERDAS IMAGINE (14.0),
- MATLAB (v. R2016a),
- ArcGIS (v. 10.3.1) ve
- Microsoft Excel

programları kullanılmıştır.

Kullanılan yazılımlardan MATLAB ve Microsoft Excel programları, öğrenci hesabı oluşturularak temin edilmiştir. ERDAS IMAGINE ve ArcGIS yazılımlarının ise deneme sürümleri kullanılmıştır.

Donanım olarak Lenovo Z500 model, 64-bit işlemci, 2 GB NVIDIA GeForce GT 740M ekran kartı, 8 GB RAM kapasite 2.10 GHz hız özelliklerine sahip bilgisayar kullanılmıştır.

#### 4.3 Uygulamada Kullanılan Veriler ve Özellikleri

Uygulamada Karadeniz bölgesine ait 5 adet, Terkos bölgesine ait 4 adet, İzmir bölgesine ait 1 adet, Antalya bölgesine ait 1 adet ve Erçek Gölü' ne ait 1 adet 30 metre mekânsal çözünürlüklü, 16-bit radyometrik çözünürlüklü LANDSAT 8 uydu görüntüleri kullanılmıştır.

LANDSAT 8 uydusu 11 Şubat 2013 yılında fırlatılmıştır. LANDSAT 8, elektromanyetik spektrumda farklı frekans aralıklarını ölçmektedir. Bu frekans aralıkları bant olarak isimlendirilmektedir ve LANDSAT 8 uydusunun 11 adet bandı bulunmaktadır. Bu bantların elektromanyetik spektrum aralıkları ve çözünürlükleri Çizelge 4. 1' de sunulmuştur [61].

Çizelge 4. 1 LANDSAT 8 uydusuna ait bantların spektral aralıkları ve çözünürlükleri

Bant Numarası	$\mu\text{m}$ (mikrometre)	Çözünürlük (metre)
1	0.433 – 0.453	30
2	0.450 – 0.515	30
3	0.525 – 0.600	30
4	0.630 – 0.680	30
5	0.845 – 0.885	30
6	1.560 – 1.660	30
7	2.100 – 2.300	30
8	0.500 – 0.680	15
9	1.360 – 1.390	30
10	10.6 – 11.2	100
11	11.5 – 12.5	100

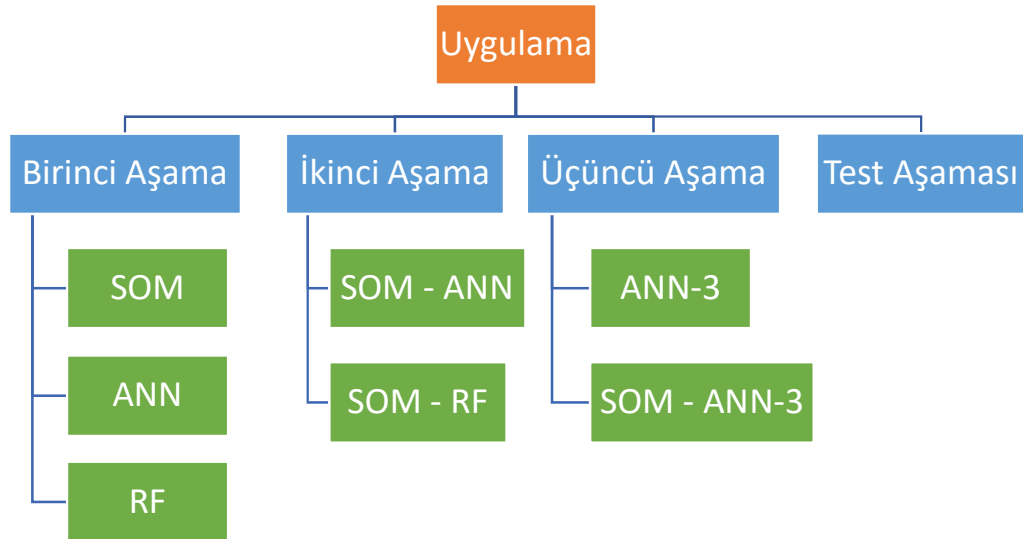
Kullanılan LANDSAT 8 uydu görüntülerinin tarihleri Çizelge 4. 2' de gösterilmiştir.

Çizelge 4. 2 Uygulamada kullanılan uydu görüntülerinin tarihleri

Veri Türü	Bölge	Görüntü Numarası	Görüntü Tarihi
Eğitim	KARADENİZ	1	8 Eylül 2017
		2	10 Eylül 2017
		3	13 Eylül 2017
		4	19 Eylül 2017
		5	20 Eylül 2017
Test	TERKOS	6	11 Eylül 2017
		7	13 Ekim 2017
		8	6 Eylül 2015
	İZMİR ANTALYA ERÇEK GÖLÜ	9	30 Temmuz 2013
		10	27 Eylül 2017
		11	9 Kasım 2017
		12	5 Eylül 2017

#### 4.4 Uygulama İşlem Adımları

Bu bölümde uygulama için kullanılacak görüntüler hazırlanmış, 3 aşamada yöntemlerin uygulamaları gerçekleştirilmiş ve son aşama olan test aşamasında, belirlenen en iyi ANN konfigürasyonu test görüntülerine uygulanarak sonuçlar gözlemlenmiştir.



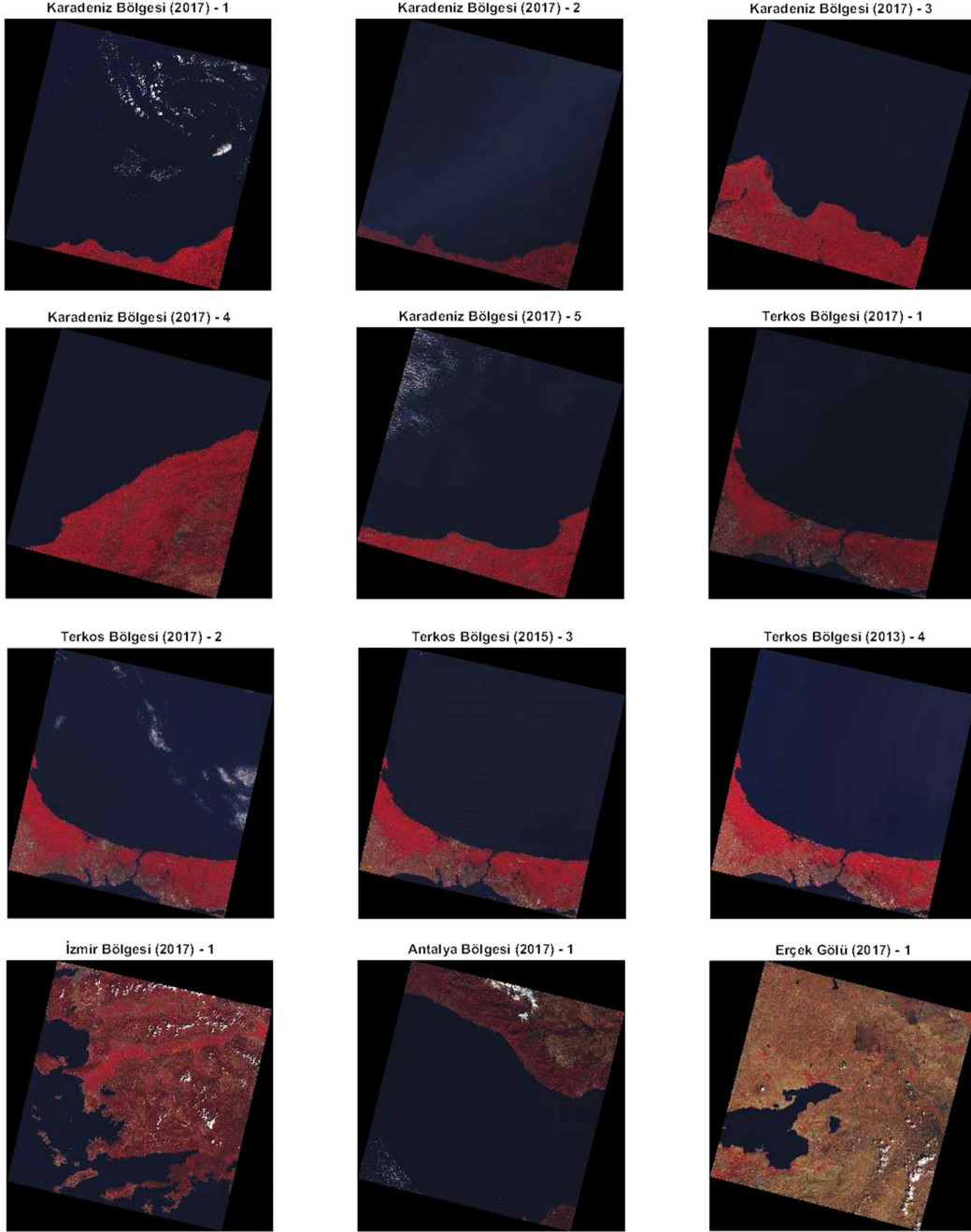
Şekil 4. 5 Uygulama işlem adımları

Şekil 4. 5' te uygulama aşamaları ve bu aşamalarda hangi işlemlerin yapılacağı kısaltmalar ile gösterilmiştir. Birinci aşamada yer alan SOM kısaltması kendini düzenleyen haritalar yöntemi uygulamasını, ANN kullanıcı tarafından toplanan eğitim verilerinin kullanıldığı yapay sinir ağları yöntemi uygulamasını ve RF ise yine kullanıcı tarafından toplanan eğitim verilerinin kullanıldığı rastgele orman yöntemi uygulamasını temsil etmektedir. İkinci aşamada yer alan SOM – ANN kısaltması SOM yöntemi sonuçları kullanılarak rastgele toplanan eğitim verilerinin kullanıldığı yapay sinir ağları yöntemi uygulamasını ve SOM – RF yine SOM yöntemi sonuçları kullanılarak rastgele toplanan eğitim verilerinin kullanıldığı rastgele orman yöntemi uygulamasını temsil etmektedir. Birinci ve ikinci aşamada, ANN yöntemi için en yüksek doğruluk sağlayan eğitim ve aktivasyon fonksiyonları belirlenmiştir. Üçüncü aşamada ise bu eğitim ve aktivasyon fonksiyonları kullanılarak en yüksek doğruluk sağlayan gizli katman ve iterasyon sayıları belirlenmiştir. ANN-3 kısaltması kullanıcı tarafından toplanan eğitim verilerinin kullanıldığı yapay sinir ağları yöntemi uygulamasını ve SOM – ANN-3 kısaltması SOM yöntemi sonuçları kullanılarak rastgele toplanan eğitim verilerinin kullanıldığı yapay sinir ağları yöntemi uygulamasını temsil etmektedir.

#### **4.4.1 Kullanılacak Görüntülerin Hazırlanması**

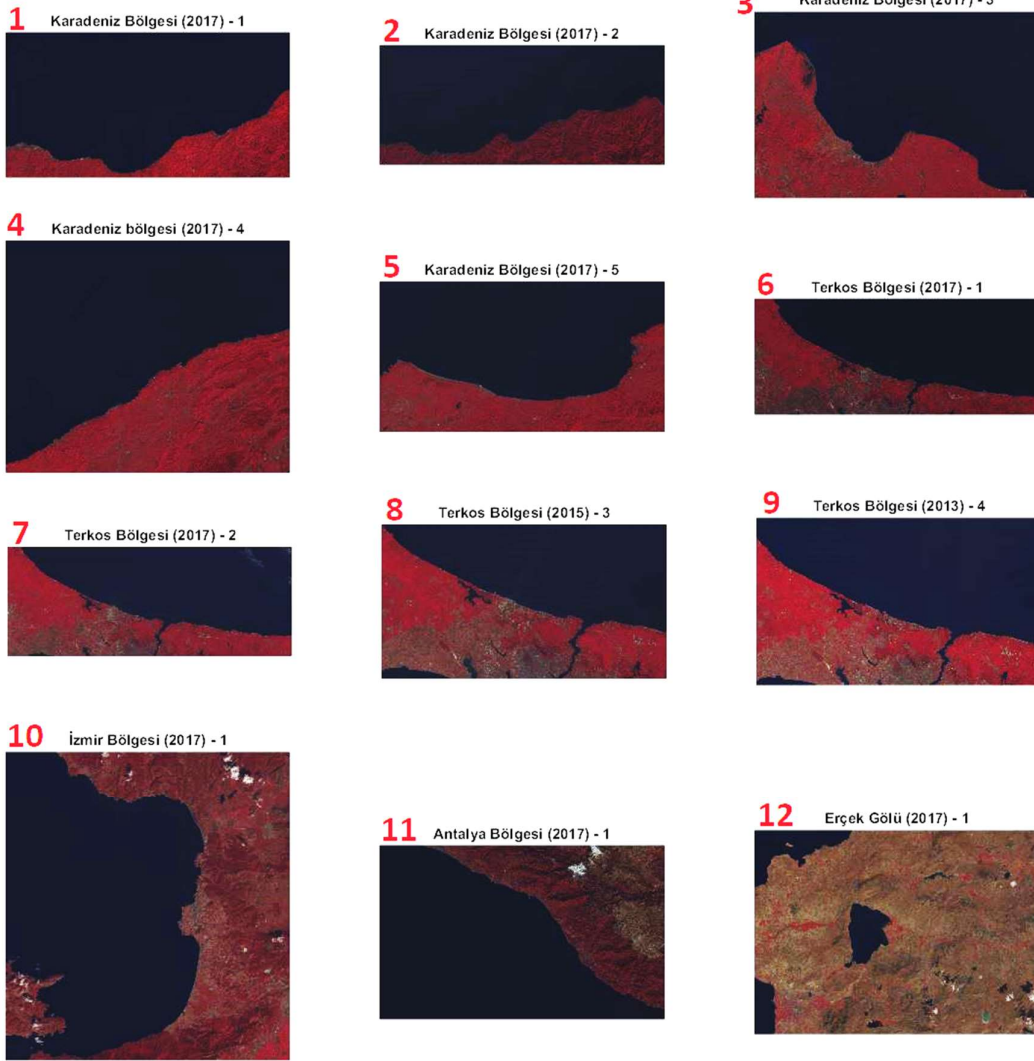
İndirilen LANDSAT 8 uydu görüntülerine ait yakın kızılötesi, kırmızı ve mavi (NIR-RED-BLUE) bantları ERDAS programı ile birleştirilmiş ve 3 bantlı görüntüler elde edilmiştir.





Şekil 4. 6 Uygulamada kullanılan uydu görüntülerine ait NIR-RED-BLUE kombinasyonları

Yapılan bu uygulamada donanım yetersizliğinden dolayı görüntüler kırılmış ve Şekil 4.7’ de gösterilmiştir. Uygulamada bu kırılan görüntüler kullanılmıştır.



Şekil 4. 7 Uygulamada kullanılan kırılmış uydu görüntülerine ait NIR-RED-BLE kombinasyonları

Bu görüntülerin boyutları Çizelge 4. 3’ te sunulmuştur.

Çizelge 4. 3 Kullanılan görüntülerin satır ve sütun sayıları

Bölge	Görüntü Numarası	Satır	Sütun
KARADENİZ	1	1682	3274
	2	1250	2963
	3	2855	4338
	4	3274	4001
	5	2059	3878
TERKOS	6	2209	5745
	7	2438	5942
	8	2130	3927
	9	2381	4036
İZMİR	10	1588	1250
ANTALYA	11	2947	4850
ERÇEK GÖLÜ	12	1691	2369

#### 4.4.2 Birinci Aşama

Uygulamanın birinci aşamasında ilk önce Terkos bölgesine ait Çizelge 4. 3' te yer alan 6 numaralı görüntü kara ve deniz olacak şekilde SOM yöntemiyle kümelendi. Daha sonra Karadeniz bölgesine ait 5 farklı uydu görüntüsünden kullanıcı tarafından toplanan eğitim verileri ile yapay sinir ağları eğitilmiş ve bu ağlar 6 numaralı görüntü ile test edilmiştir. RF yönteminde ağaç yapılarını oluşturmak için yapay sinir ağlarının eğitiminde kullanılan aynı veriler kullanılarak ağaç yapıları oluşturulmuştur. Bu ağaç yapıları da 6 numaralı görüntü ile test edilmiştir.

SOM yönteminde verilerin normalize edilmesi gerekmediği için orijinal 16-bit değerleri kullanılmıştır.

MATLAB programında bulunan **selforgmap** fonksiyonu ile SOM ağları tasarlanmıştır. Bu fonksiyonun ihtiyaç duyduğu parametreler Çizelge 4. 4'de sunulmuştur [62].

Çizelge 4. 4 SOM parametreleri

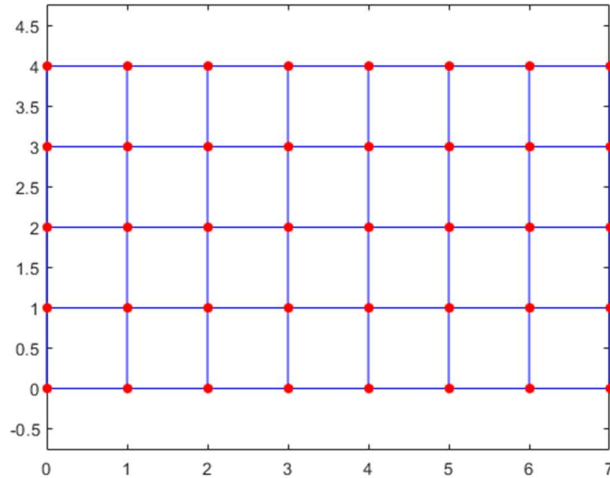
dimensions	Oluşturulacak haritanın boyutları
coverSteps	Öğrenme iterasyon sayısı
initNeighbor	Komşuluk boyutu
topologyFcn	Topoloji Fonksiyonu
distanceFcn	Mesafe Fonksiyonu

Bu uygulamada, dimensions (harita boyutları), coverSteps (öğrenme iterasyon sayısı), initNeighbor (komşuluk boyutu) ve epok sayısı (eğitim iterasyon sayısı) sırasıyla [2 1], 100, 3 ve 2000 olarak alınmıştır. Çıktı görüntünün iki sınıflı olabilmesi için harita boyutları [2 1] seçilmiştir. Eğitim iterasyon sayısı ise 2000 seçilerek, eğitimin tamamlanamaması ortadan kaldırılmaya çalışılmıştır. Öğrenme iterasyon sayısı ve komşuluk boyutu ise MATLAB programında varsayılan olarak gelmektedir ve bu çalışmada değiştirilmemiştir. topologyFcn (topoloji fonksiyonu) ile distanceFcn (mesafe fonksiyonu) parametreleri ile oluşturulabilecek kombinasyonlar üzerinde durulmuştur.

#### 4.4.2.1 SOM Yöntemine Ait Topoloji Fonksiyonları

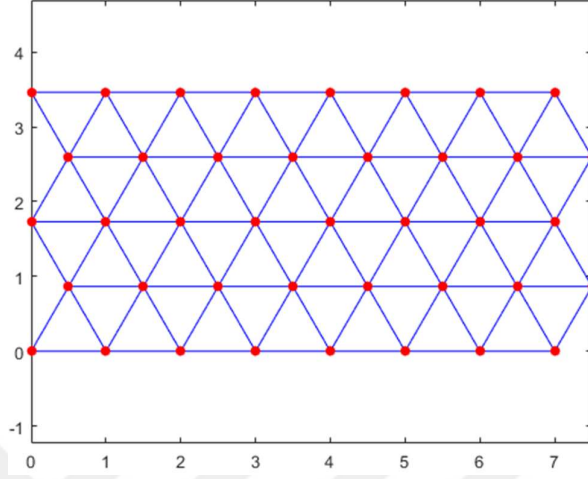
MATLAB programında kullanılabilir 3 farklı topoloji fonksiyonu bulunmaktadır. Bunlar sırasıyla grid ağı topolojisi (gridtop), altıgen ağ topolojisi (hextop) ve rastgele ağ topolojisi (randtop) şeklindedir [63].

**Grid Ağ Topolojisi:** Çıktı harita üzerindeki nöronlar ızgara ağı oluşturacak şekilde dizilmektedir.



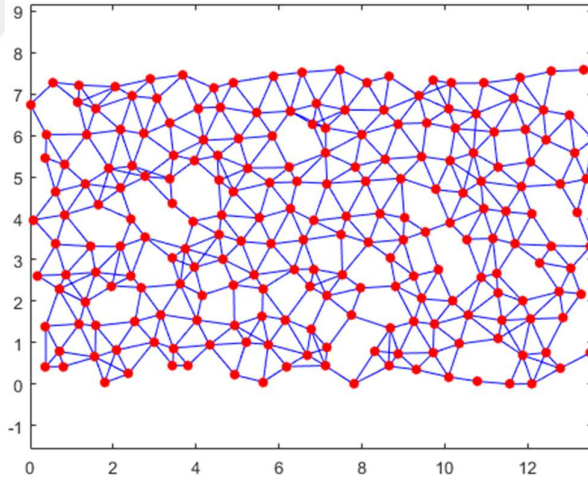
Şekil 4. 8 Grid ağ topolojisine göre konumlandırılmış nöronlar

**Altıgen Ağ Topolojisi:** Çıktı harita üzerindeki nöronlar altıgen ağ oluşturacak şekilde dizilmektedir.



Şekil 4. 9 Altıgen ağ topolojisine göre konumlandırılmış nöronlar

**Rastgele Ağ Topolojisi:** Çıktı harita üzerindeki nöronlar rastgele konumlandırılarak ağ oluşturulmaktadır.



Şekil 4. 10 Rastgele ağ topolojisine göre konumlandırılmış nöronlar

#### 4.4.2.2 SOM Yöntemine Ait Mesafe Fonksiyonları

MATLAB programında kullanılabilen 4 farklı mesafe fonksiyonu bulunmaktadır. Bunlar sırasıyla öklid mesafe fonksiyonu (dist), kutu mesafe fonksiyonu (boxdist), bağlantı mesafe fonksiyonu (linkdist) ve manhattan mesafe fonksiyonu (mandist) şeklindedir [43].

**Öklid Mesafe Fonksiyonu:** X ve Y 2 nöronu temsil eden 2 vektör olmak üzere, bu 2 nöron arasındaki mesafe olan öklid mesafesi

$$d_{dist} = \text{sqr}t(\text{sum}(X - Y)^2) \quad (4.1)$$

denklemleri ile bulunmaktadır [44].

**Kutu Mesafe Fonksiyonu:** X ve Y 2 nöronu temsil eden 2 vektör olmak üzere, bu 2 nöron arasındaki mesafe olan kutu mesafesi

$$d_{boxdist} = \text{max}(\text{abs}(X - Y)) \quad (4.2)$$

denklemleri ile bulunmaktadır [44].

**Bağlantı Mesafe Fonksiyonu:** Nöronlar arasındaki öklid mesafeleri 1' den küçük veya eşit ise, bağlantı mesafesi 1 olarak belirlenmektedir. Aralarındaki mesafe ölçülen nöronlara eşit ve 1 birim uzaklıkta N tane nöron var ise, bağlantı mesafesi N alınır. Bu olasılıkların dışında kalan durumlarda ise, bağlantı mesafesi nöron sayısına eşit alınır [43].

**Manhattan Mesafe Fonksiyonu:** X ve Y 2 nöronu temsil eden 2 vektör olmak üzere, bu 2 nöron arasındaki mesafe olan manhattan mesafesi

$$d_{mandist} = \text{sum}(\text{abs}(X - Y)) \quad (4.3)$$

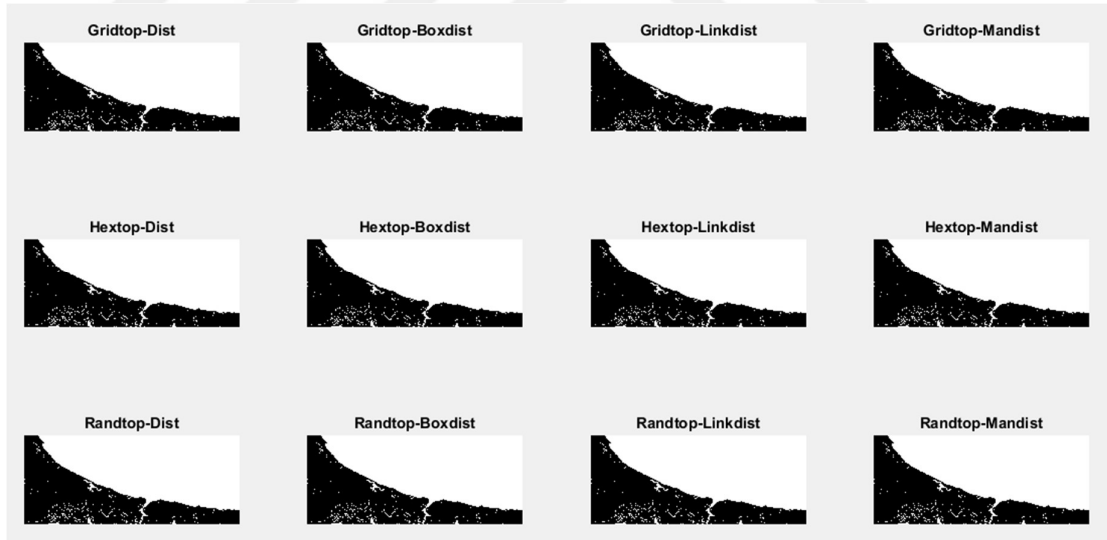
denklemleri ile bulunmaktadır [44].

Topoloji ve mesafe fonksiyonları kullanılarak oluşturulabilecek SOM ağı kombinasyonları Çizelge 4. 5' te gösterilmiştir. Bu kombinasyonların her biri için ayrı ayrı SOM ağı tasarlanmıştır.

Çizelge 4. 5 SOM kombinasyonları

Kombinasyon Numarası	Topoloji Fonksiyonu	Mesafe Fonksiyonu
1	Gridtop	Dist
2	Gridtop	Boxdist
3	Gridtop	Linkdist
4	Gridtop	Mandist
5	Hextop	Dist
6	Hextop	Boxdist
7	Hextop	Linkdist
8	Hextop	Mandist
9	Randtop	Dist
10	Randtop	Boxdist
11	Randtop	Linkdist
12	Randtop	Mandist

Terkos bölgesine ait 6 numaralı görüntü üzerinde Çizelge 4. 5'teki 12 kombinasyonun her biri ayrı ayrı denenmiştir ve kara-deniz olacak şekilde ikili görüntüler elde edilmiştir.



Şekil 4. 11 Kombinasyonlara ait ikili sonuç görüntüler

Çizelge 4. 6' da sonuçların elde edilme süreleri yer almaktadır.

Çizelge 4. 6 Kombinasyon sonuçlarının elde edilme süreleri

<b>Kombinasyon</b>	<b>Zaman</b>
Gridtop – Dist	1 saat 55 dakika
Gridtop – Boxdist	1 saat 48 dakika
Gridtop – Linkdist	1 saat 52 dakika
Gridtop – Mandist	1 saat 49 dakika
Hextop – Dist	1 saat 46 dakika
Hextop – Boxdist	1 saat 48 dakika
Hextop – Linkdist	1 saat 46 dakika
Hextop – Mandist	1 saat 46 dakika
Randtop – Dist	1 saat 48 dakika
Randtop – Boxdist	1 saat 46 dakika
Randtop – Linkdist	1 saat 50 dakika
Randtop – Mandist	1 saat 48 dakika

Çizelge 4. 6' daki sonuçlar incelendiğinde, bütün kombinasyonlar yaklaşık olarak aynı sürede sonuca ulaşmıştır. En kısa ve en uzun sürede sonuca ulaşan kombinasyonlar renklendirilmiştir. Sonuç ikili görüntülerin hepsinden kıyı çizgisi çıkartılmış ve 6 numaralı görüntü üzerinde elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Karşılaştırma sonuçlarından elde edilen ortalama hataların metre ve piksel karşılıkları Çizelge 4. 7' de gösterilmiştir.



Çizelge 4. 7 Kombinasyonlara ait kara ve deniz piksel sayıları

<b>Kombinayon</b>	<b>Ortalama Hata (metre)</b>	<b>Ortalama Hata (piksel)</b>
Gridtop – Dist	14,61	0,49
Gridtop – Boxdist	14,61	0,49
Gridtop – Linkdist	14,61	0,49
Gridtop – Mandist	14,61	0,49
Hextop – Dist	14,61	0,49
Hextop – Boxdist	14,61	0,49
Hextop – Linkdist	14,61	0,49
Hextop – Mandist	14,61	0,49
Randtop – Dist	14,61	0,49
Randtop – Boxdist	14,61	0,49
Randtop – Linkdist	14,61	0,49
Randtop – Mandist	14,61	0,49

Çizelge 4. 7' de de görüldüğü gibi hesaplanan ortalama hataların hepsi aynı sonuçlar ortaya koymuştur.

Şekil 4. 7' deki 9 numaralı görüntü için de Çizelge 4. 5' deki kombinasyonların hepsi denenmiştir. Elde edilen sonuç ikili görüntülerin hepsinden kıyı çizgisi çıkartılmış ve 9 numaralı görüntü üzerinde elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Bütün kombinasyonlar için hesaplanan ortalama hataların hepsi 14,17 m / 0,47 piksel sonucunu ortaya koymuştur.

Elde edilen sonuçlar aynı çıktığından dolayı, Kohonen tarafından önerilen Hextop-Dist kombinasyonu seçilmiştir. Kohonen' e göre altıgen ağ topolojisi ve öklid mesafe fonksiyonu kullanıldığında daha tutarlı ve doğru sonuçlar alındığı görülmüştür [64]. Bu nedenle ikinci aşamada Karadeniz bölgesine ait görüntülere SOM yöntemi uygulanırken altıgen ağ topolojisi ve öklid mesafe fonksiyonu kullanılmıştır.

ANN yöntemi uygulaması da MATLAB ortamında gerçekleştirilmiştir. Tasarlanan ağlara çıktı olarak sadece kara ve deniz sınıfları üretecek şekilde karadan ve denizden toplanan eğitim verileri sunulmuştur. ANN yönteminin uygulanabilmesi için verilerin normalize edilmesi gerekmektedir. Bu nedenle eğitim için kullanılan ve test için kullanılan verileri maksimum-minimum normalizasyon yöntemine göre 0-1 aralığına normalize edilmiştir. Maksimum-minimum normalizasyonu,

$$x'_i = \frac{x_i - x_{minimum}}{x_{maksimum} - x_{minimum}} \quad (4.4)$$

denklemleri kullanılarak bir  $x$  veri setindeki bütün  $x_i$  değerleri 0-1 aralığına normalize edilmektedir [65].

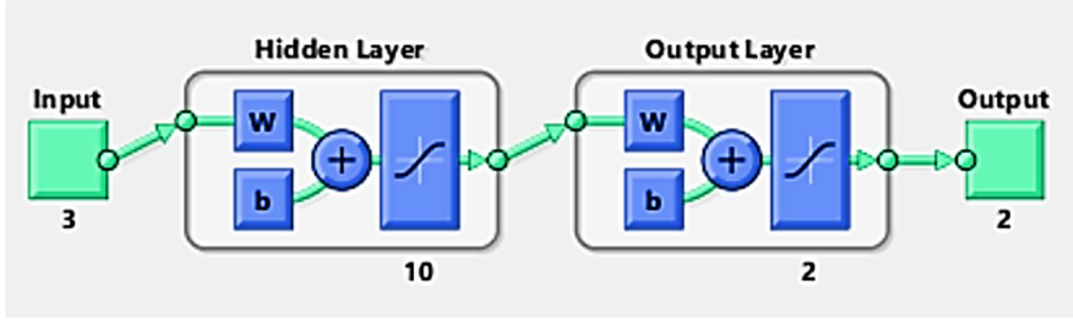
Kullanılan 3 bantlı verilerin her bandına ayrı ayrı maksimum-minimum normalizasyonu uygulanmıştır.

Ağların hepsi ileri beslemeli ve geri yayımlıdır. İlk olarak yapay sinir ağlarında gizli katman sayısı 1 ve bu katmandaki nöron sayısı 10, iterasyon sayısı da 2000 olarak belirlenmiştir. 2 eğitim fonksiyonu ile 2 aktivasyon fonksiyonları arasında oluşan kombinasyonlar test edilmiş olup seçilen eğitim fonksiyonları Levenber – Marquardt (TRAINLM) ile Ölçekli Conjugate Gradient (TRAINSCLG), aktivasyon fonksiyonları ise Hyperbolic Tangent Sigmoid (TANSIG) ve Logistic Sigmoid (LOGSIG) şeklindedir. Bu fonksiyonların oluşturduğu kombinasyonlar Çizelge 4. 8' de gösterilmiştir.

Çizelge 4. 8 Tasarlanan ağ kombinasyonları

Kombinasyon	Eğitim Fonksiyonu	Aktivasyon Fonksiyonu
1	TRAINLM	TANSIG
2	TRAINLM	LOGSIG
3	TRAINSCLG	TANSIG
4	TRAINSCLG	LOGSIG

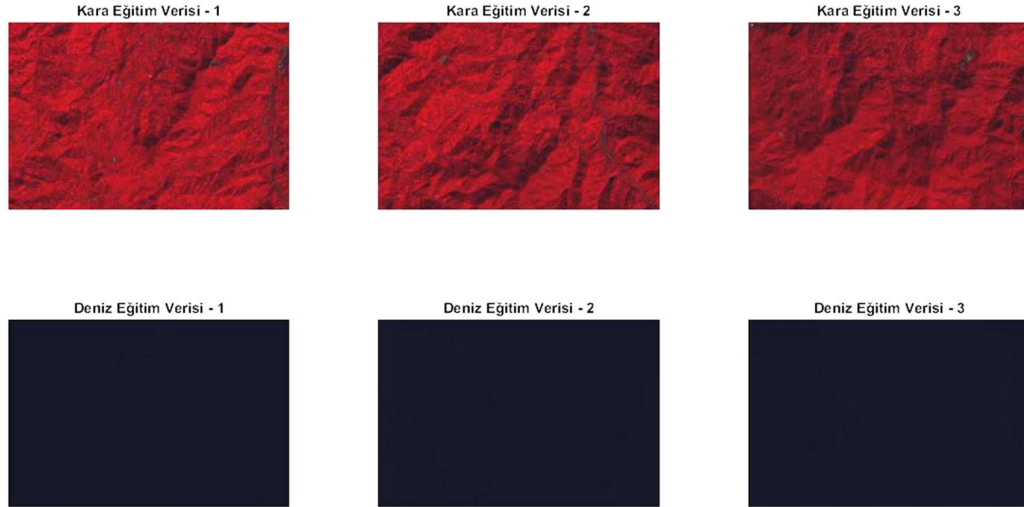
Çizelge 4. 8' deki kombinasyonlar ile oluşturulmuş yapay sinir ağı mimarileri Şekil 4. 12' de gösterilmiştir.



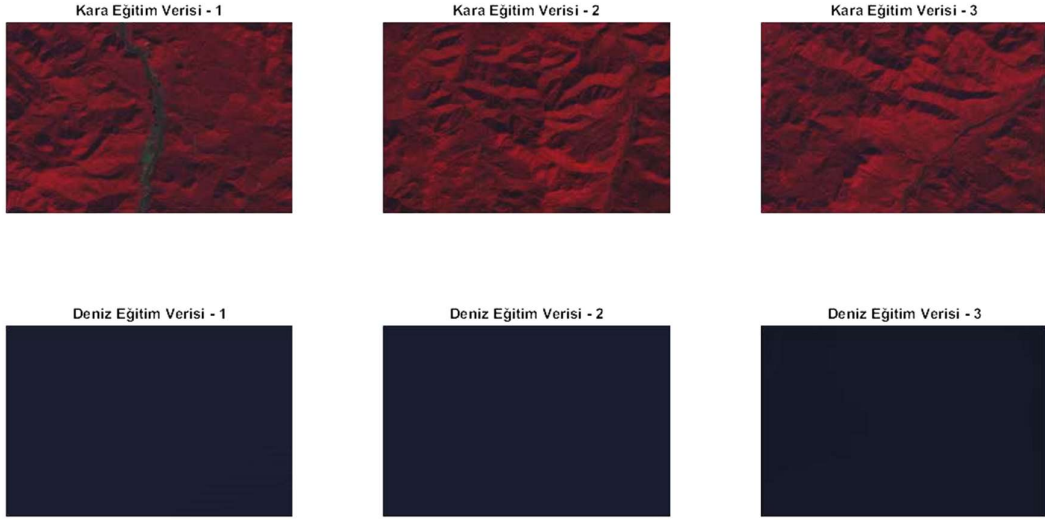
Şekil 4. 12 Kombinasyonlardaki tek gizli katmanlı ağ tasarımı

Şekil 4. 12' deki Input girdi görüntülerin 3 bandını, Hidden Layer ağdaki 10 nöronlu gizli katmanı, Output Layer gizli katmanlardan gelen ağırlıklandırılmış verilerin toplandığı katmanı ve Output ise ağın çıktısını temsil etmektedir.

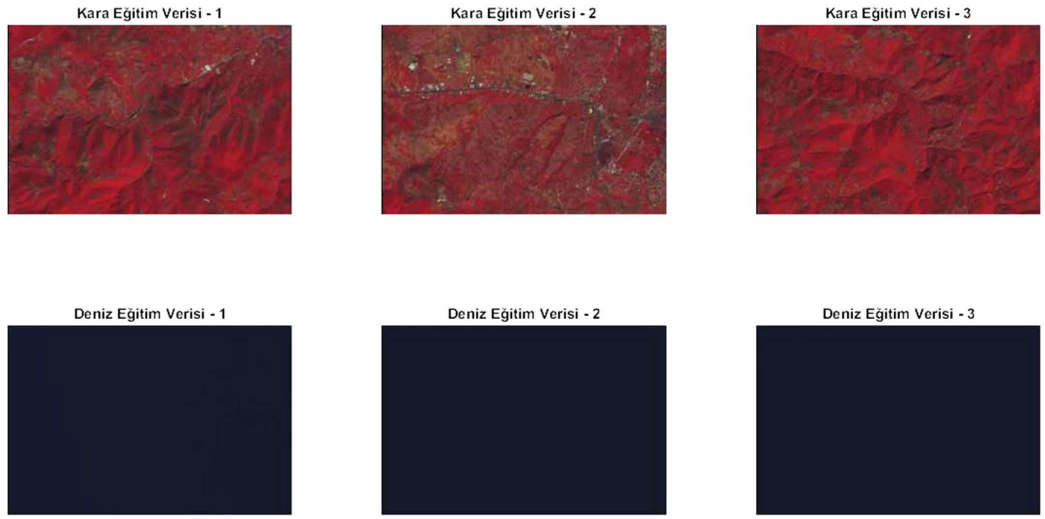
Karadeniz bölgesine ait 5 uydu görüntüsünün her birinden 175000 karadan ve 175000 denizden olmak üzere 350000 piksel veri toplanmıştır. Toplamda 5 görüntü kullanılarak karadan 875000 ve denizden 875000 olmak üzere 1750000 piksel ağların eğitimi için kullanılmıştır. Toplanan bu eğitim verilerinin miktarı kullanılan donanım ve yazılımın yeterliliğine göre belirlenmiştir.



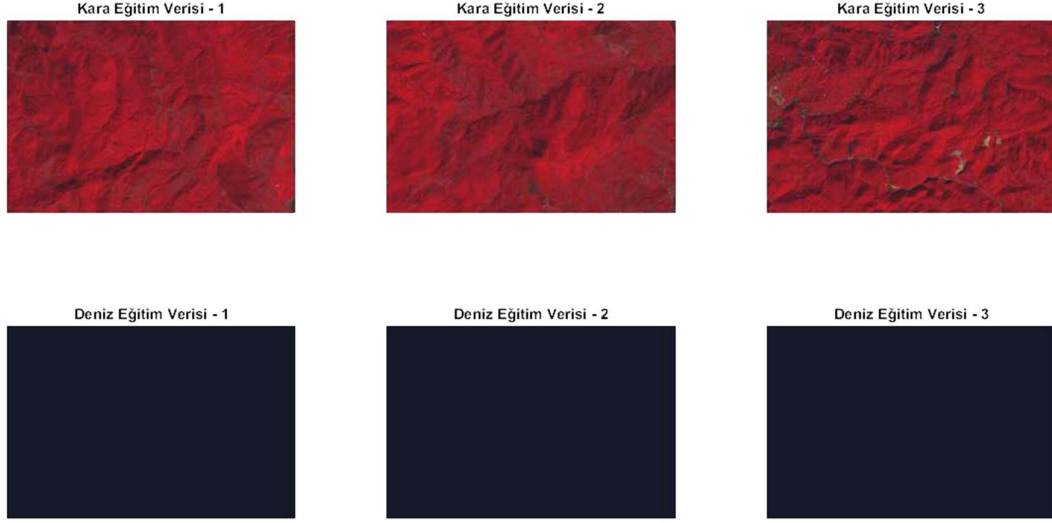
Şekil 4. 13 1 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)



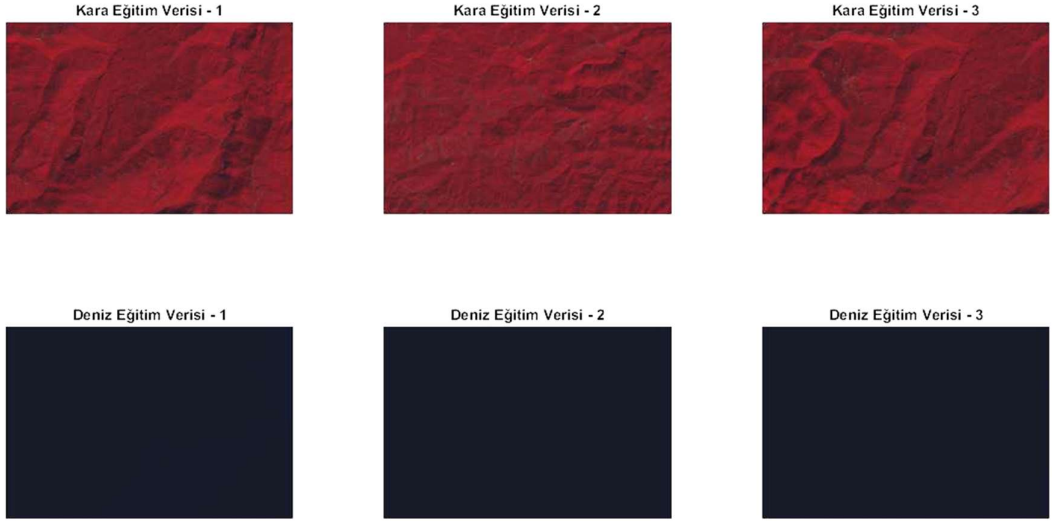
Şekil 4. 14 2 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)



Şekil 4. 15 3 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)



Şekil 4. 16 4 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)



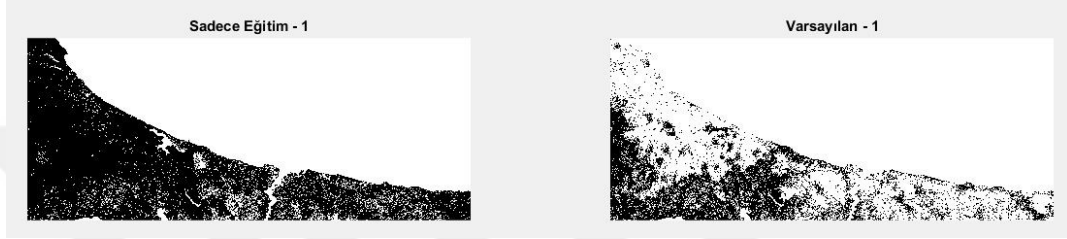
Şekil 4. 17 5 numaralı görüntüden toplanan eğitim verileri (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)

Şekil 4. 13 ve Şekil 4. 17 arası şekiller yapay sinir ağlarının eğitiminde kullanılan Karadeniz bölgesine ait 5 farklı uydu görüntüsünden toplanmış verileri göstermektedir.

Tasarlanan ağların eğitilmesi 2 farklı şekilde gerçekleştirilmiştir. Bunların birincisi girdi eğitim verisinin tamamının eğitim için kullanılarak eğitilmesi, ikincisi ise girdi eğitim verisinin %70' inin eğitim için, %15 'inin test için ve %15 'inin ise doğrulama için kullanılarak eğitilmesidir. Bu yüzdelerle giren veriler eğitim verisi içerisinde rastgele belirlenmektedir [66, 67].

Birinci eğitim şekli kısaca 'Sadece Eğitim', ikinci eğitim şekli ise kısaca 'Varsayılan' olarak isimlendirilmiştir. Kullanıcı tarafından toplanan eğitim verileri ile eğitilen ağlar kısaca ANN ağları olarak isimlendirilmiştir.

Bu eğitim şekilleri dikkate alınarak her bir kombinasyon için 2 farklı ağ tasarlanmıştır. En iyi sonucu veren ağın belirlenmesi için Terkos bölgesine ait 6 numaralı görüntü kullanılmıştır.



Şekil 4. 18 TRAINLM – TANSIG kombinasyonu sonuç görüntüleri (ANN ağı)



Şekil 4. 19 TRAINLM – LOGSIG kombinasyonu sonuç görüntüleri (ANN ağı)



Şekil 4. 20 TRAINSCG – TANSIG kombinasyonu sonuç görüntüleri (ANN ağı)



Şekil 4. 21 TRAINSCG – LOGSIG kombinasyonu sonuç görüntüleri (ANN ağı)

Şekil 4. 18 – Şekil 4. 21 arası şekillerde ANN ağ kombinasyonlarından elde edilen sonuçlar yer almaktadır.

ANN ağlarının eğitilme süreleri Çizelge 4. 9’ da gösterilmiştir.

Çizelge 4. 9 ANN ağlarının eğitilme süreleri

<b>Kombinasyon</b>	<b>Sadece Eğitim</b>	<b>Varsayılan</b>
TRAINLM – TANSIG	12 dakika	10 dakika
TRAINLM – LOGSIG	18 dakika	12 dakika
TRAINSÇG – TANSIG	48 dakika	8 dakika
TRAINSÇG – LOGSIG	22 dakika	3 dakika

Çizelge 4. 9 ‘da gösterilen kombinasyonların eğitilme süreleri incelendiğinde, en hızlı eğitim TRAINSÇG – LOGSIG (Varsayılan) kombinasyonunda ve en yavaş eğitim ise TRAINSÇG – TANSIG (Sadece Eğitim) kombinasyonunda gerçekleşmiştir. ANN ağlarında kullanılan eğitim verilerinin ağ çıktıları ile hedef çıktı verileri karşılaştırılarak ortalama karesel hatalar hesaplanmış ve Çizelge 4. 10’ da gösterilmiştir.

Çizelge 4. 10 ANN ağlarının ortalama karesel hataları

<b>Kombinasyon</b>	<b>Sadece Eğitim</b>	<b>Varsayılan</b>
TRAINLM – TANSIG	2,36e-04	2,34e-04
TRAINLM – LOGSIG	2,38e-04	2,21e-04
TRAINSÇG – TANSIG	2,51e-04	4,55e-04
TRAINSÇG – LOGSIG	2,78e-04	7,01e-04

Çizelge 4. 10’ da verilen ortalama karesel hatalar incelendiğinde, bütün kombinasyonların yaklaşık aynı değere sahip olduğu görülmektedir.

ANN ağlarında kullanılan eğitim verilerine ait ağ çıktılarının gradyan değerleri Çizelge 4. 11’ de sunulmuştur.

Çizelge 4. 11 ANN ağlarının gradyan değerleri

<b>Kombinasyon</b>	<b>Sadece Eğitim</b>	<b>Varsayılan</b>
TRAINLM – TANSIG	3,93e-08	2,66e-04
TRAINLM – LOGSIG	9,66e-08	2,00e-06
TRAINSÇG – TANSIG	1,90e-05	4,18e-05
TRAINSÇG – LOGSIG	9,91e-07	1,25e-04

Çizelge 4. 11' de yer alan gradyan değerlerine bakıldığında, en düşük değer TRAINLM – TANSIG (Sadece Eğitim) kombinasyonu ile elde edilirken, en büyük değer ise TRAINLM – TANSIG (Varsayılan) kombinasyonu ile elde edilmiştir. Gradyan değerinin küçük olması, ağların eğitimlerini tamamlaması için bir kriterdir.

Bu ağlardan elde edilen sonuç görüntüleri incelendiğinde, kara ile deniz sınıfının birbirinden ayrılabilirdiği yalnızca 5 görüntü vardır ve yalnızca bunlardan kıyı çizgisi çıkarılabilmektedir. Ardından elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Geriye kalan 3 görüntüden kara ve deniz sınıfları net bir şekilde ayrılamadığı için karşılaştırma yapılamamıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Hesaplanan ortalama hata değerleri Çizelge 4. 12' de sunulmuştur.

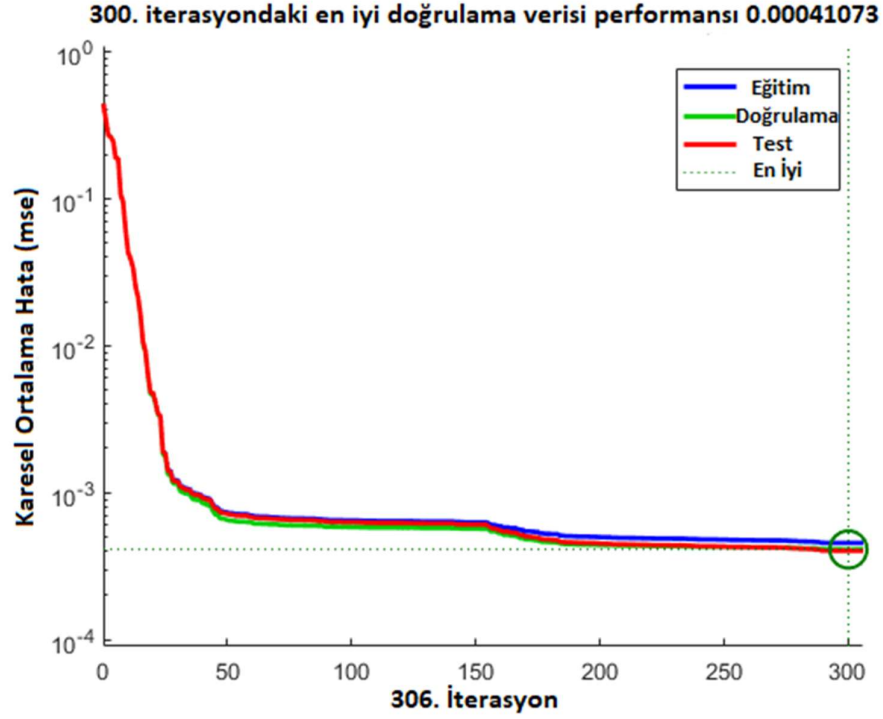
Çizelge 4. 12 ANN ağlarına ait karşılaştırma sonuçlarının ortalama hataları

Kombinasyon	Ortalama Hata (Sadece Eğitim)		Ortalama Hata (Varsayılan)	
	metre	piksel	metre	piksel
TRAINLM – TANSIG	32,21	1,07	-	-
TRAINLM – LOGSIG	32,84	1,09	29,37	0,98
TRAINSÇG – TANSIG	-	-	18,31	0,61
TRAINSÇG – LOGSIG	-	-	21,08	0,70

Çizelge 4. 12, ANN ağlarına ait sonuç görüntülerden elde edilen kıyı çizgilerinin ortalama hata değerlerini göstermektedir. Çizelge 4. 12' de yer alan metre/piksel bazlı ortalama hata değerleri incelendiğinde, en iyi sonuç TRAINSÇG – TANSIG (Varsayılan) kombinasyonu ile ve en kötü sonuç ise TRAINLM – LOGSIG (Sadece Eğitim) kombinasyonu ile elde edilmiştir.

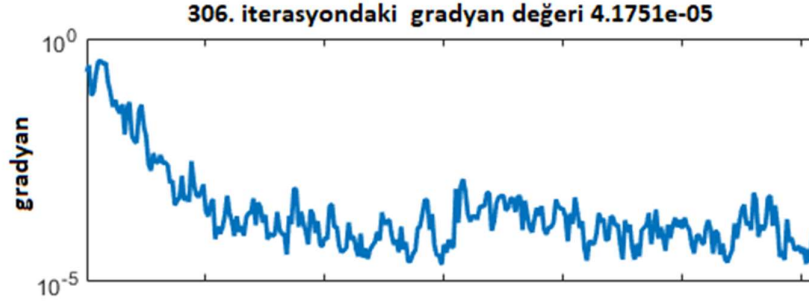
En iyi sonucu veren TRAINSÇG – TANSIG (Varsayılan) kombinasyonuna ait performans grafiği Şekil 4. 22' de sunulmuştur.





Şekil 4. 22 TRAINSCG – TANSIG (Varsayılan) kombinasyonu performans grafiği

Şekil 4. 22’deki grafikte dikey sütun ortalama karesel hatayı, yatay sütun ise iterasyon sayısını temsil etmektedir. Mavi çizgi eğitim verisine ait ortalama karesel hatayı, yeşil çizgi doğrulama verisine ilişkin ortalama karesel hatayı ve kırmızı çizgi ise test verisinin ortalama karesel hatasını temsil etmektedir. Şekil 4. 22’de, eğitim, doğrulama ve test verilerinin karesel ortalama hatalarının ilk başta hızlı bir şekilde azaldığı gözlemlenmiştir. Daha sonra ise küçük değişimler ile en iyi sonuca ulaşmıştır. Tasarlanan bu ağ 300. iterasyondaki performansını 306. iterasyona kadar 6 defa test etmiştir ve değişim olmadığı için eğitimi durdurmuştur. MATLAB programında 6 defa test etme durumu sabittir fakat bu değiştirilebilmektedir. Sunulan tez çalışması kapsamında tasarlanan ağlarda bu değer sabit kabul edilip değiştirilmemiştir.



Şekil 4. 23 TRAINSCG – TANSIG (Varsayılan) kombinasyonu gradyan deęerleri grafięi

Varsayılan eęitim şekli kullanılarak tasarlanan aęlar, eęitimlerini iterasyon sayısını tamamlamadan doęrulama verisinin karesel ortalama hatasına veya eęitim verisinin gradyan deęerine gre erkenden tamamlayabilmektedir. Gradyan deęerine gre eęitimin erken tamamlanması iin TRAINLM eęitim fonksiyonu kullanıldıęında aęın gradyan deęeri  $1,00e-06$  deęerinden, TRAINSCG eęitim fonksiyonu kullanıldıęında  $1,00e-07$  deęerinden daha dşk olması gerekmektedir. Bu deęerler MATLAB programında sabit parametre olarak gelmektedir, ama bu deęerleri deęiřtirmek mmkndr. Sunulan tez alıřmasında tasarlanan aęlarda bu deęerler deęiřtirilmemiřtir. Şekil 4. 22’ de performans grafięi yer alan aę doęrulama verisinin en iyi performansına gre eęitimini tamamlamıřtır. Ayrıca bu aęın iterasyon sayısına baęlı olarak gradyan deęerlerindeki deęiřimi gsteren grafik Şekil 4. 23’ de sunulmuřtur. Gradyan deęerleri, her iterasyonda aędan ıkan tahmini deęerlerin gradyan deęerleridir ve belirlenen gradyan deęerine doęru azalarak devam eder.

RF yntemi de MATLAB ortamında gerekleřtirilmiřtir. Burada aęa yapılarını oluřturmak iin ANN aęlarında kullanılan eęitim verileri kullanılmıřtır. RF ynteminde, verilerin normalize edilmesi gerekmedięi iin orijinal 16-bit deęerleri kullanılmıřtır. 50, 250 ve 500 aęalı yapılar oluřturulmuřtur.

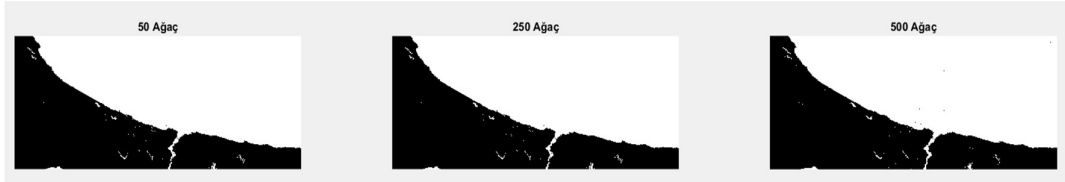
ANN aęları iin kullanılan eęitim verisi kullanılarak oluřturulan bu aęa yapıları kısaca RF aęa yapıları olarak isimlendirilmiřtir.

RF yntemi aęa sayısı (N) ve aęaların en iyi řekilde dallanmasını belirleyen (m) parametrelerinden oluřmaktadır.

Breiman (2001) yapmış olduđu çalışmada, m sayısının deđişken sayısının karekök karřılıđı olarak alındığında genel olarak optimum sonuca en yakın sonuç elde edildiđini belirtmiřtir [68].

Sunulan tez çalışmasında, uydu görüntülerinin yalnızca 3 bandı kullanıldıđı için  $\sqrt{3} = 1,73$  deđeri elde edilmektedir. Bu deđere en yakın tam sayı 2 olduđu için bu parametre 2 olarak seçilmiřtir.

RF ağaç yapıları oluşturulduktan sonra en iyi sonucu veren ağaç yapısının belirlenmesi için Terkos bölgesine ait 6 numaralı görüntü kullanılmıřtır. Elde edilen kıyı çizgileri elle yapılan sayısallařtırma sonuçları ile karřılařtırılmıřtır.



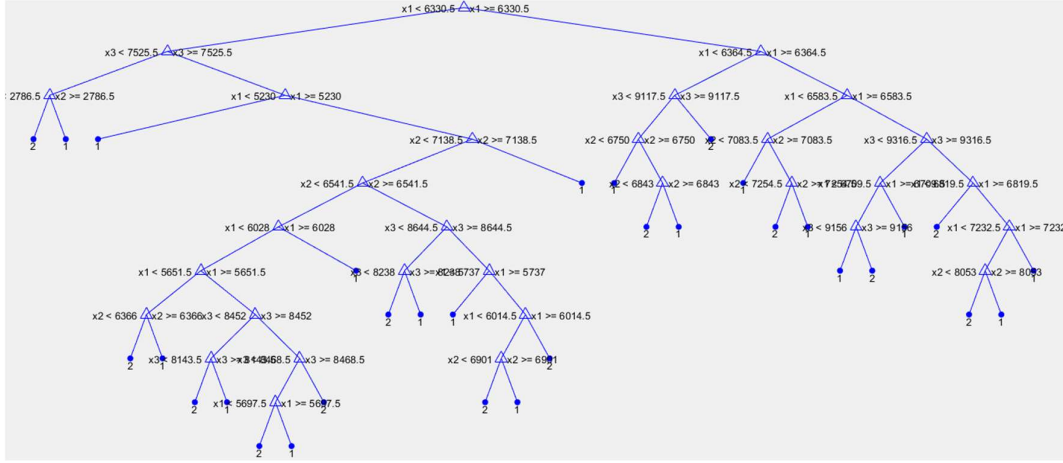
řekil 4. 24 Sırasıyla 50-250-500 ağaç sayılı yapılar için sonuç sınıflandırılmıř görüntüler (RF ağaç yapıları)

RF ağaç yapılarının oluşturulma süreleri Çizelge 4. 13' te sunulmuřtur.

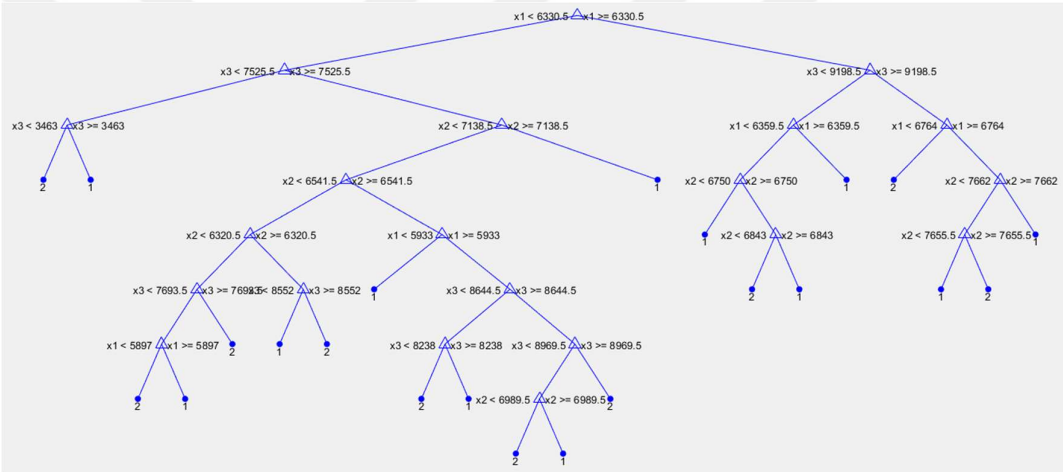
Çizelge 4. 13 Görüntülerin elde edilme süreleri

Geçen Zaman		
50 Ağaç	250 Ağaç	500 Ağaç
19 dakika	59 dakika	111 dakika

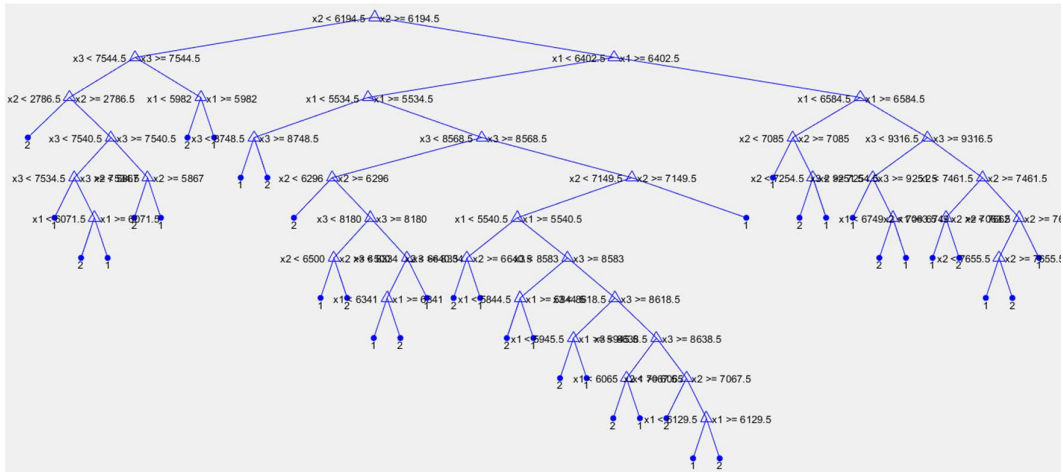
Çizelge 4. 13' teki sonuçlar incelendiğinde sonuca en hızlı 50 ağaçlı ağaç yapısı ulařırken, sonuca en yavaş 500 ağaçlı ağaç yapısı ulařmıřtır. Bu bilgilere bakılarak ağaç sayısının artması, sonuca ulařma süresini negatif yönde etkilediđi söylenebilmektedir.



Şekil 4. 25 50 ağaç sayılı ağaç yapısının 50. Ağacına ait dallanmalar



Şekil 4. 26 250 ağaç sayılı ağaç yapısının 50. Ağacına ait dallanmalar



Şekil 4. 27 500 ağaç sayılı ağaç yapısının 50. Ağacına ait dallanmalar

Şekil 4. 25 ve Şekil 4. 27 arası şekillerde RF ağaç yapılarının 50. ağaçları gösterilmiştir. Şekil 4. 24' deki görüntülerden elde edilen kıyı çizgileri, elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Hesaplanan ortalama hatalar Çizelge 4. 14' te gösterilmiştir.

Çizelge 4. 14 Sonuç görüntülere ait karşılaştırma sonuçları

Ağaç Sayısı	Karşılaştırma Sonuçları	
	Ortalama Hata (metre)	Ortalama Hata (piksel)
50 Ağaç	13,64	0,45
250 Ağaç	14,26	0,48
500 Ağaç	15,39	0,51

Çizelge 4. 14' deki sonuçlar incelendiğinde, 50 ağaca sahip ağaç yapısının en iyi sonucu verdiği gözlemlenmektedir. En kötü sonucu ise 500 ağaca sahip ağaç yapısı vermiştir. Çizelge 4. 14' teki metre/piksel bazlı ortalama hata değerlerine göre, 50 ağaçtan oluşan RF ağaç yapısının 250 ve 500 ağaç sayılı RF ağaç yapılarına göre daha iyi sonuç ortaya koyduğu gözlemlenmiştir.

#### 4.4.3 İkinci Aşama

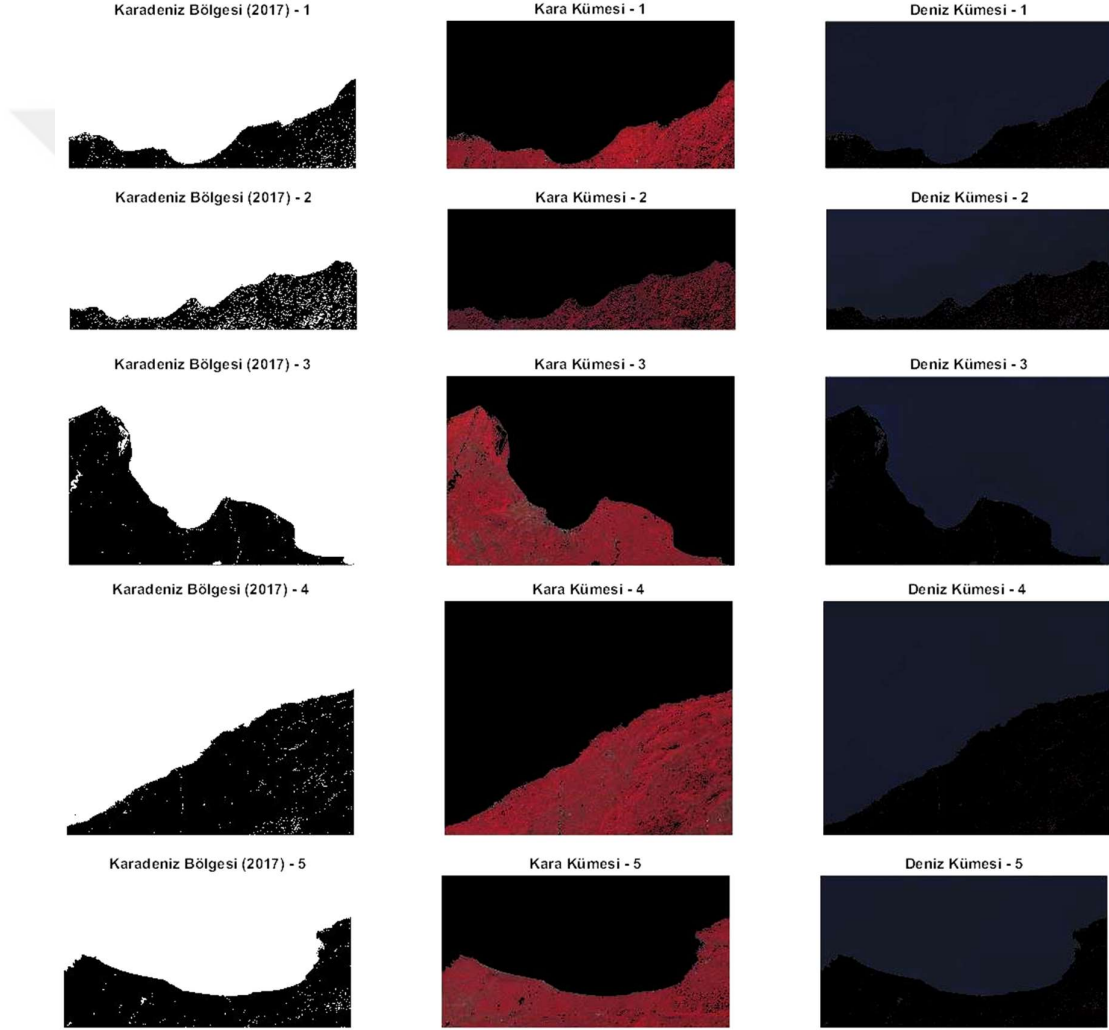
Uygulamanın ikinci aşamasında SOM yöntemi Karadeniz bölgesine ait 1, 2, 3, 4, 5 numaralı görüntülere uygulanmıştır. Daha sonra ANN ağlarına ait kara ve deniz eğitim verileri ile aynı sayıda olacak şekilde SOM sonuçları kullanılarak rastgele eğitim verileri toplanmıştır. Çizelge 4. 8' deki kombinasyonları ile yeni yapay sinir ağları oluşturulmuş ve bu ağlar rastgele toplanan eğitim verileri ile eğitilmiştir.

Rastgele toplanan eğitim verileri ile eğitilen ağlar, kısaca SOM – ANN ağları olarak isimlendirilmiştir.

En iyi sonucu veren SOM – ANN ağının belirlenmesi için Terkos bölgesine ait 6 numaralı görüntü kullanılmıştır.

RF yöntemi için SOM – ANN ağları için kullanılan eğitim verileri kullanılarak ağaç yapıları oluşturulmuş ve en iyi sonucu veren ağaç yapısının belirlenmesi için Terkos bölgesine ait 6 numaralı görüntü kullanılmıştır.

Uygulanan SOM yönteminde Kohonen tarafından önerilen altıgen topoloji fonksiyonu ile öklid mesafe fonksiyonundan oluşan kombinasyon kullanılarak deniz ve kara kümeleri oluşturulmuştur.



Şekil 4. 28 SOM yöntemi sonuçları (1. sütun SOM yöntemi sonuçları, 2. sütun kara kümeleri, 3. sütun deniz kümeleri)

Şekil 4. 28' de Karadeniz bölgesine ait 1, 2, 3, 4 ve 5 numaralı görüntülerin SOM yöntemi uygulaması sonucu oluşan iki kümeli görüntüleri ve bu ikili görüntülerin maske olarak kullanılması sonucu elde edilen kara-deniz kümelerine ait görüntüler yer almaktadır.

Şekil 4. 28' deki görüntülerin elde edilme süreleri ise Çizelge 4. 15' de gösterilmiştir.

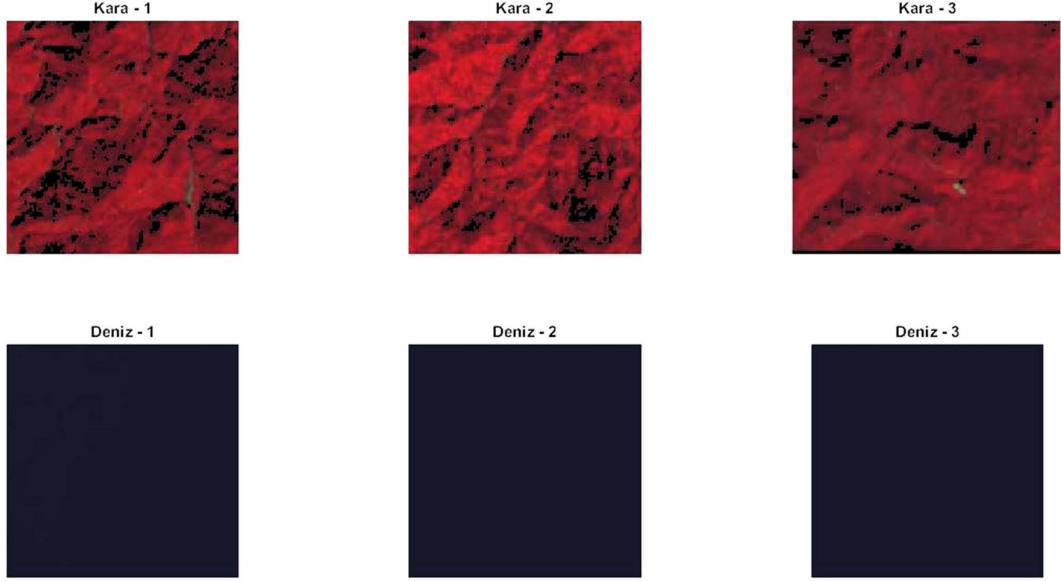
Çizelge 4. 15 Karadeniz bölgesine ait görüntülerin elde edilme süreleri

Bölge	Görüntü Numarası	Zaman
Karadeniz	1	50 dakika
	2	40 dakika
	3	1 saat 37 dakika
	4	1 saat 46 dakika
	5	1 saat 11 dakika

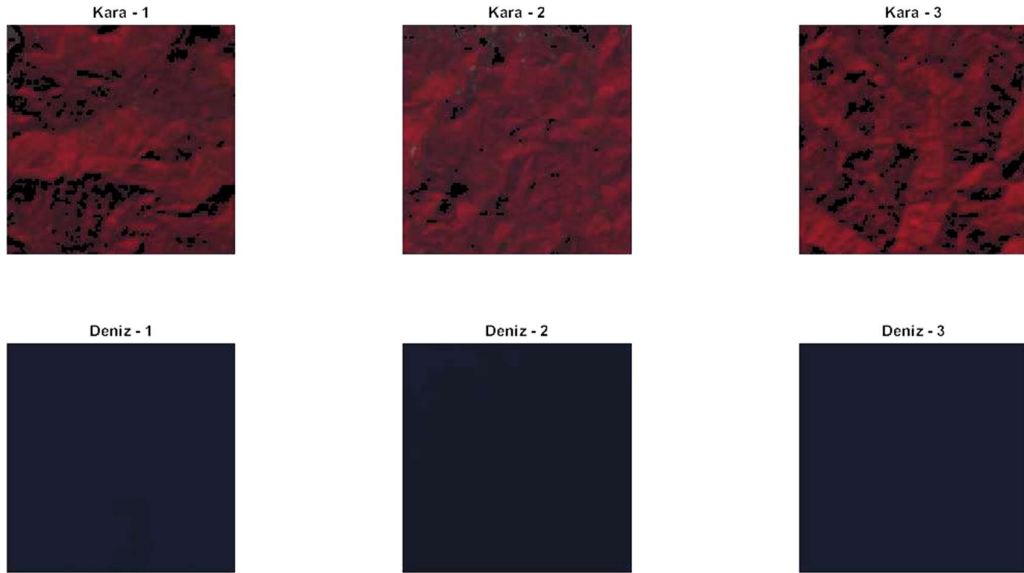
SOM yöntemiyle kümelenmiş Şekil 4. 28' deki 5 görüntüden rastgele 4 numaralı görüntü seçilip kıyı çizgisi çıkartılmıştır. Daha sonra elle yapılan sayısallaştırmasıyla karşılaştırılmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Bu karşılaştırma sonucuna göre SOM yöntemiyle kıyı çizgisi 23,06 m/0,77 piksel hata ile bulunmuştur.

İkinci aşamada da ANN ağları kombinasyonları kullanılarak yeni ağlar tasarlanmıştır. Birinci aşamadan farklı olarak bu ağların eğitimi SOM yöntemi sonuçlarına göre rastgele toplanan eğitim verileri ile yapılmıştır. SOM yöntemi sonucu oluşan kara ve deniz kümelerinden rastgele eğitim verileri toplanması işlemi MATLAB ortamında yazılan bir program ile gerçekleştirilmiştir. Bu program ile kümelenmiş görüntülerden otomatik olarak rastgele görüntü parçaları kırılmış ve kırılan görüntüler eğitim verisi için kullanılmıştır. Rastgele otomatik görüntü kırma işlemi yapan bu program üç adımda çalışmaktadır. İlk adımda, rastgele veri toplanmak istenen deniz veya kara kümesini içeren görüntü yüklenmektedir. İkinci adımda, yüklenen görüntü üzerinden otomatik olarak rastgele pikseller seçilmektedir. Üçüncü adımda ise, seçilen bu pikseller, kırılacak görüntülerin sol üst köşeleri olacak şekilde ayarlanmaktadır ve kullanıcı tarafından belirlenen satır-sütun değerlerine göre görüntüler kırılmaktadır. Rastgele seçilen piksellerin kenarlara gelmesi durumunda, kullanıcı tarafından belirlenen satır-sütun sayısında görüntüler oluşturulamamaktadır. Fakat kullanıcının belirlediği boyutlardan daha düşük ve rastgele seçilen pikselin konumuna göre değişen boyutta görüntüler oluşturulmaktadır.

Sunulan bu tez çalışmasında, her bir kara ve deniz kümesine ait görüntüden bu program yardımıyla 100 satır ve 100 sütundan oluşan 100 adet toplamda 1000 adet görüntü rastgele toplanmıştır.

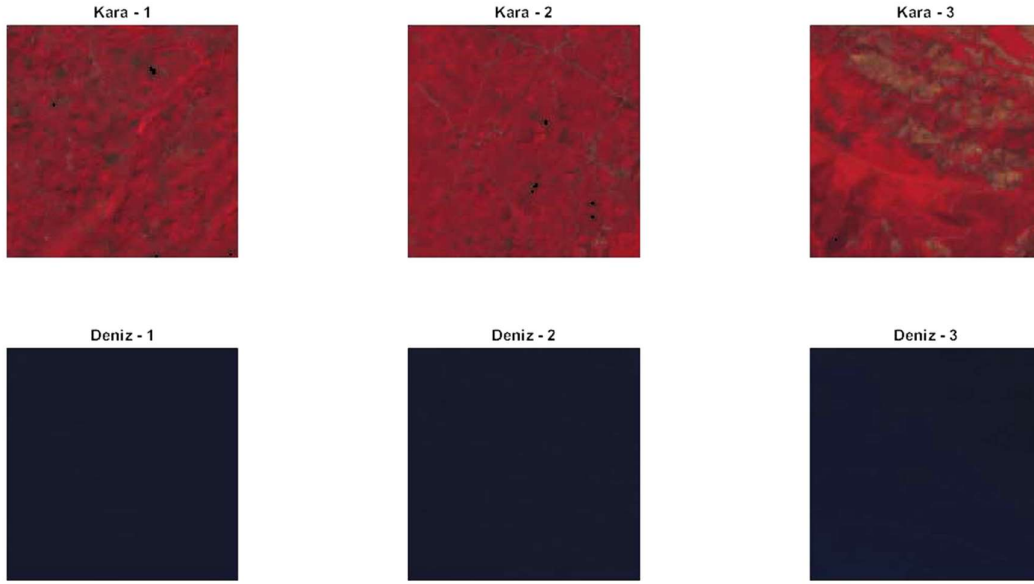


Şekil 4. 29 1 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)

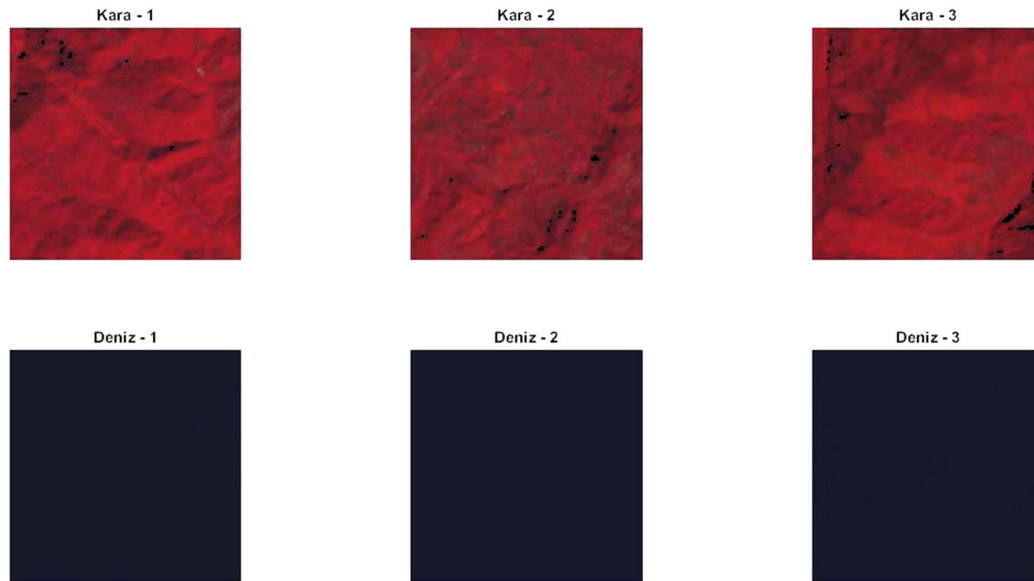


Şekil 4. 30 2 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)

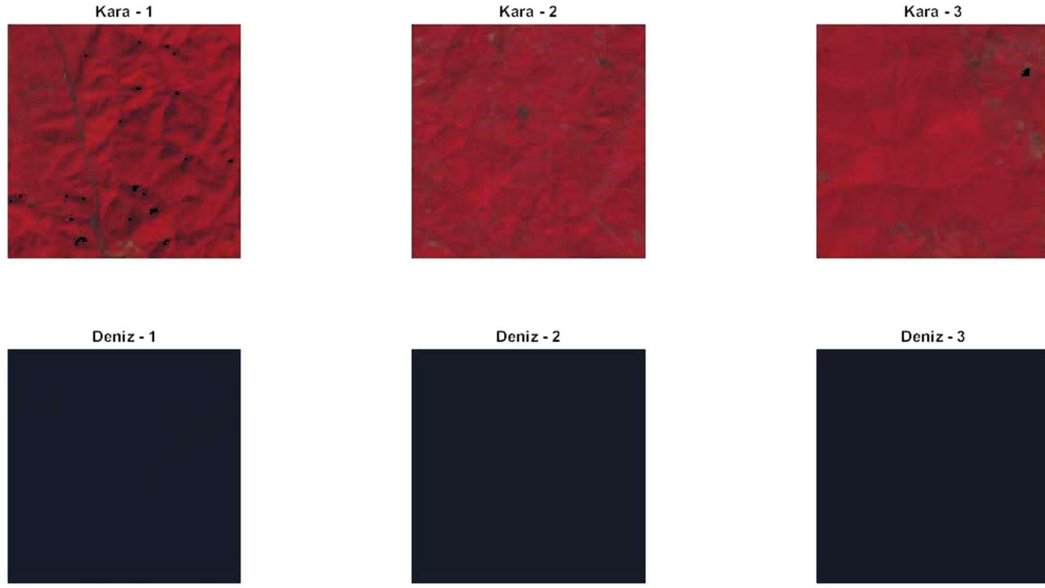




Şekil 4. 31 3 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)



Şekil 4. 32 4 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)



Şekil 4. 33 5 numaralı görüntünün SOM yöntemi sonucuna göre rastgele toplanan görüntülerden örnekler (üstteki 3 görüntü kara, alttaki 3 görüntü deniz)

Şekil 4. 29 ve Şekil 4. 33 arası şekillerde, SOM yöntemi sonucu rastgele toplanmış görüntülerden örnekler gösterilmiştir.

Rastgele toplanmış bu görüntülerden 1, 2, 3, 4 ve 5 numaralı her bir görüntü için 175000 adet piksel karadan ve 175000 adet piksel denizden otomatik olarak seçilmiştir. Toplamda 5 görüntü için rastgele toplanan verilerden 875000 kara pikseli ve 875000 deniz pikseli olmak üzere toplamda 1750000 piksel seçilmiştir. ANN ağlarına ait eğitim verileri ile aynı sayıda eğitim verisi SOM sonuçları kullanılarak otomatik olarak toplanmıştır.

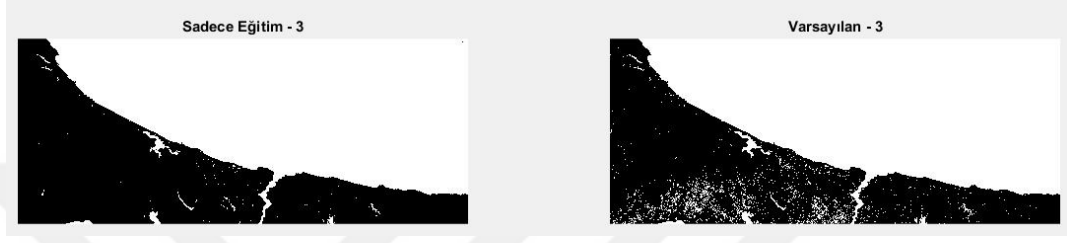
ANN ağları için kullanılan Sadece Eğitim e Varsayılan eğitim şekilleri, SOM – ANN ağları için de kullanılarak ağ tasarımları gerçekleştirilmiştir. En iyi sonucu veren ağın belirlenmesi için Terkos bölgesine ait 6 numaralı görüntü kullanılmıştır.



Şekil 4. 34 TRAINLM – TANSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı)



Şekil 4. 35 TRAINLM – LOGSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı)



Şekil 4. 36 TRAINSCG – TANSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı)



Şekil 4. 37 TRAINSCG – LOGSIG kombinasyonu sonuç görüntüleri (SOM-ANN ağı)

SOM – ANN ağlarının eğitilme süreleri Çizelge 4. 16' da gösterilmiştir.

Çizelge 4. 16 SOM – ANN ağlarının eğitilme süreleri

Kombinasyon	Sadece Eğitim	Varsayılan
TRAINLM – TANSIG	1 saat 8 dakika	18 dakika
TRAINLM – LOGSIG	1 saat 22 dakika	11 dakika
TRAINSCG – TANSIG	25 dakika	2 dakika
TRAINSCG – LOGSIG	49 dakika	3 dakika

Çizelge 4. 16' da gösterilen kombinasyonların eğitilme süreleri incelendiğinde, en hızlı eğitim TRAINSCG – TANSIG (Varsayılan) kombinasyonunda ve en yavaş eğitim ise TRAINLM – LOGSIG (Sadece Eğitim) kombinasyonunda gerçekleşmiştir.

SOM – ANN ağlarında kullanılan eğitim verilerinin ağ çıktıları ile hedef çıktı verileri karşılaştırılarak ortalama karesel hatalar hesaplanmış ve Çizelge 4. 17' de gösterilmiştir.

Çizelge 4. 17 SOM – ANN ağlarının ortalama karesel hataları

<b>Kombinasyon</b>	<b>Sadece Eğitim</b>	<b>Varsayılan</b>
TRAINLM – TANSIG	0,00123	0,00132
TRAINLM – LOGSIG	0,00131	0,00145
TRAINSÇG – TANSIG	0,00341	0,00375
TRAINSÇG – LOGSIG	0,00328	0,00358

Çizelge 4. 17’ de verilen ortalama karesel hatalar incelendiğinde, TRAINLM – TANSIG ve TRAINLM – LOGSIG kombinasyonlarının yaklaşık aynı ortalama karesel hataya sahip olduğu görülmektedir. Aynı şekilde TRAINSÇG – TANSIG ve TRAINSÇG – LOGSIG kombinasyonlarının da yaklaşık aynı ortalama karesel hataya sahip olduğu gözlemlenmiştir.

SOM – ANN ağlarında kullanılan eğitim verilerine ait ağ çıktılarının gradyan değerleri Çizelge 4. 18’ de sunulmuştur.

Çizelge 4. 18 SOM – ANN ağlarının gradyan değerleri

<b>Kombinasyon</b>	<b>Sadece Eğitim</b>	<b>Varsayılan</b>
TRAINLM – TANSIG	9,32e-08	1,77e-05
TRAINLM – LOGSIG	5,20e-08	4,78e-05
TRAINSÇG – TANSIG	9,82e-07	2,82e-04
TRAINSÇG – LOGSIG	6,13e-06	7,54e-05

Çizelge 4. 18’ de yer alan gradyan değerlerine bakıldığında, en düşük değer TRAINLM – LOGSIG (Sadece Eğitim) kombinasyonu ile elde edilirken, en büyük değer ise TRAINSÇG – TANSIG (Varsayılan) kombinasyonu ile elde edilmiştir. Gradyan değerinin küçük olması, ağların eğitimlerini tamamlaması için bir kriterdir.

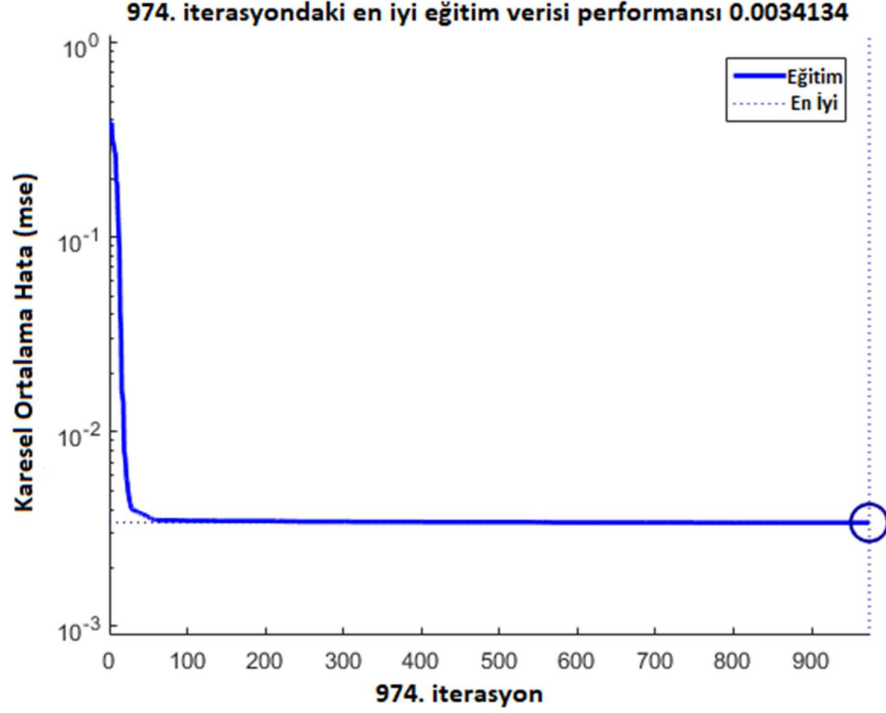
SOM – ANN ağlarından elde edilen sonuç görüntüleri incelendiğinde, 8 görüntünün hepsinde deniz ve kara sınıfı birbirinden ayrılabilmiş ve kıyı çizgileri çıkartılmıştır. Çıkartılan kıyı çizgileri elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Hesaplanan ortalama hata değerleri Çizelge 4. 19’ da sunulmuştur.

Çizelge 4. 19 SOM – ANN ağlarına ait karşılaştırma sonuçlarının ortalama hataları

Kombinasyon	Ortalama Hata (Sadece Eğitim)		Ortalama Hata (Varsayılan)	
	Metre	Piksel	Metre	Piksel
TRAINLM – TANSIG	7,50	0,25	22,11	0,74
TRAINLM – LOGSIG	13,88	0,46	11,23	0,37
TRAINSÇG – TANSIG	6,69	0,22	20,85	0,70
TRAINSÇG – LOGSIG	7,85	0,26	47,34	1,58

Çizelge 4. 19, SOM – ANN ağlarına ait sonuç görüntülerden elde edilen kıyı çizgilerinin ortalama hata değerlerini göstermektedir. Çizelge 4. 19’ da yer alan metre/piksel bazlı ortalama hata değerleri incelendiğinde, en iyi sonuç TRAINSÇG – TANSIG (Sadece Eğitim) kombinasyonu ile ve en kötü sonuç ise TRAINSÇG – LOGSIG (Varsayılan) kombinasyonu ile elde edilmiştir.

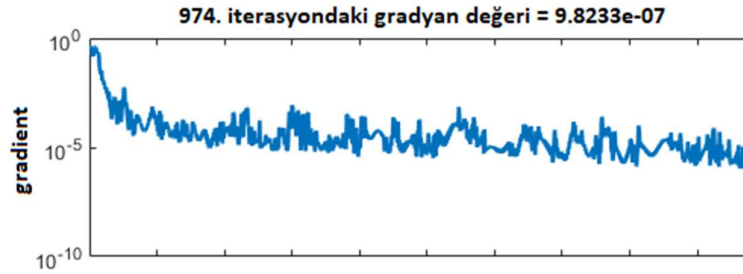
Çizelge 4. 19’ daki sonuçlar, SOM – ANN ağlarına ait ortalama hata değerlerini göstermektedir. Çizelge 4. 19’ da yer alan metre/piksel bazlı ortalama hata değerlerine göre, en iyi sonuçlar TRAINSÇG – TANSIG (Sadece Eğitim) kombinasyonu ile elde edilmiştir. Bu kombinasyona ait performans grafiği Şekil 4. 38’ de sunulmuştur.



Şekil 4. 38 TRAINSCG – TANSIG (Sadece Eğitim) kombinasyonu performans grafiği

Şekil 4. 38' deki grafikte dikey sütun ortalama karesel hatayı, yatay sütun ise iterasyon sayısını temsil etmektedir. Mavi çizgi ise eğitim veri setinin ortalama karesel hatasını göstermektedir. Şekil 4. 38' de, eğitim verisinin karesel ortalama hataları ilk başta çok hızlı bir şekilde azalmış daha sonra ise sabit kalmıştır. Tasarlanan bu ağ, hesaplanan gradyan değeri ( $9,82e-07$ ),  $1,00e-06$  sayısından daha düşük olduğu için ağ eğitimini tamamlamış ve 2000 iterasyonun tamamını yapmamıştır.

Sadece Eğitim şekli kullanılarak tasarlanan ağlar, eğitimlerini doğrulama verisi kullanılmadığından dolayı, iterasyon sayısını tamamlamadan yalnızca eğitim verisinin gradyan değerine göre erkenden tamamlayabilmektedir.

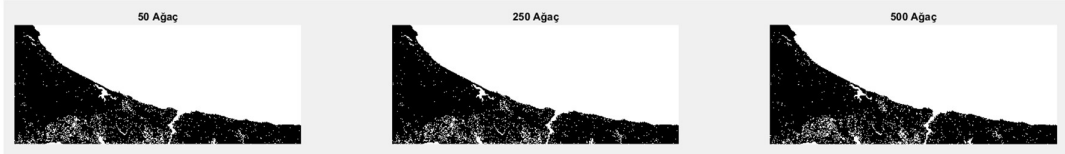


Şekil 4. 39 TRAINSCG – TANSIG (Sadece Eğitim) kombinasyonu gradyan değerleri grafiği

Şekil 4. 38' deki ağa ait her iterasyondaki gradyan değerlerinin yer aldığı grafik Şekil 4. 39'da sunulmuştur. Gradyan değerleri, her iterasyonda ağdan çıkan tahmini değerlerin gradyan değerleridir ve belirlenen gradyan değerine doğru azalarak devam eder. Bu ağ tasarımında olduğu gibi, eğer belirlenen gradyan değerine veya daha küçük bir değere ulaşırsa ağ eğitimi durdurur.

ANN ve SOM – ANN ağlarının karşılaştırma sonuçlarına göre en iyi sonucun TRAINSCG – TANSIG kombinasyonu ile elde edildiği saptanmıştır.

SOM – RF ağ yapıları 50, 250 ve 500 ağaçtan oluşmaktadır. En iyi sonucu veren ağ yapısını belirlemesi için Terkos bölgesine ait 6 numaralı görüntü kullanılmıştır.



Şekil 4. 40 Sırasıyla 50-250-500 ağaç sayılı yapılara ait sonuç sınıflandırılmış görüntüler (SOM – RF ağ yapıları)

SOM – RF ağ yapılarının oluşturulma süreleri Çizelge 4. 20' de sunulmuştur.

Çizelge 4. 20 Görüntülerin elde edilme süreleri

Geçen Zaman		
50 Ağaç	250 Ağaç	500 Ağaç
19 dakika	59 dakika	111 dakika

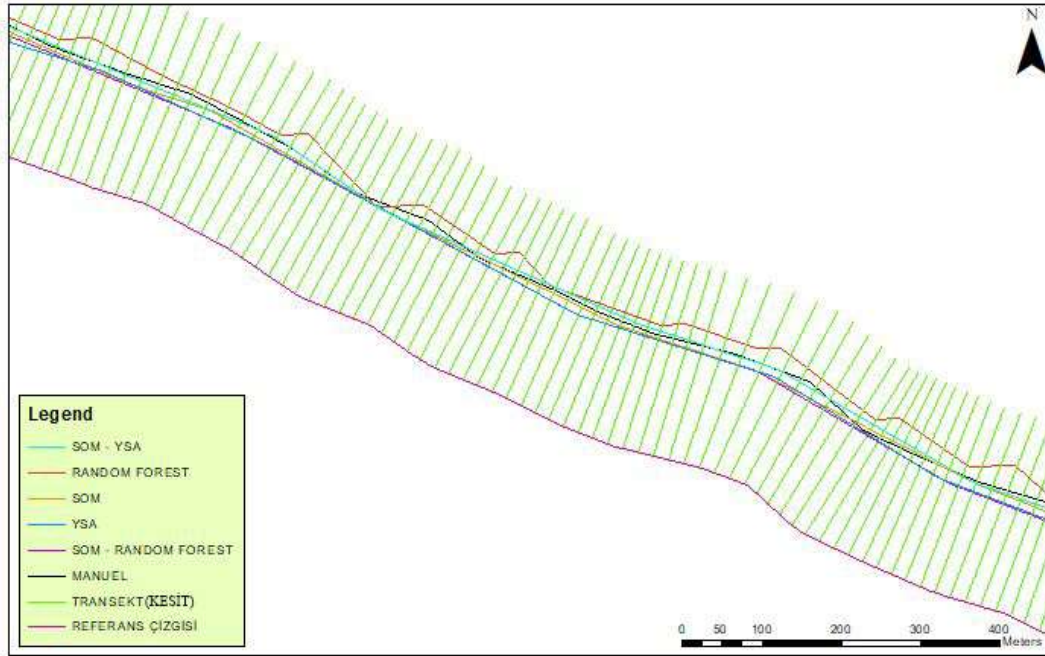
Çizelge 4. 20' deki sonuçlar incelendiğinde sonuca en hızlı 50 ağaçlı ağ yapısı ulaşırken, sonuca en yavaş 500 ağaçlı ağ yapısı ulaşmıştır. Bu bilgilere bakılarak ağaç sayısının artması, sonuca ulaşma süresini negatif yönde etkilediği söylenebilmektedir.

Şekil 4. 40' taki sonuç görüntülerden elde edilen kıyı çizgileri elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Hesaplanan ortalama hatalar Çizelge 4. 21' de bu değerler gösterilmiştir.

Çizelge 4. 21 Sonuç görüntülere ait karşılaştırma sonuçları

Ağaç Sayısı	Karşılaştırma Sonuçları	
	Ortalama Hata (metre)	Ortalama Hata (piksel)
50 Ağaç	24,28	0,81
250 Ağaç	23,69	0,79
500 Ağaç	24,03	0,80

Çizelge 4. 21’deki metre/piksel bazlı ortalama hata değerlerine göre, 250 ağaçtan oluşan SOM – RF ağaç yapısının 50 ve 500 ağaç sayılı SOM – RF ağaç yapılarına göre daha iyi sonuç ortaya koyduğu gözlemlenmiştir. Birinci aşamadaki RF ağaç yapıları ve ikinci aşamadaki SOM – RF ağaç yapılarının, metre/piksel bazlı ortalama hata değerleri karşılaştırıldığında RF ağaç yapılarına ait sonuçların daha iyi çıktığı gözlemlenmiştir.



Şekil 4. 41 Birinci ve ikinci aşamada en iyi sonucu veren kıyı çizgileri ve DSAS bileşenleri

Şekil 4. 41’da birinci ve ikinci aşamada yapılan karşılaştırma sonuçlarına göre en iyi sonuç veren kıyı çizgileri, elle yapılan sayısallaştırma, referans kıyı çizgisi ve kesitler yer almaktadır.

Şekil 4. 41’da yer alan kıyı çizgilerine ait ortalama hata değerleri Çizelge 4. 22’de gösterilmiştir.



Çizelge 4. 22 Birinci ve ikinci aşamada en iyi sonuç veren kıyı çizgilerinin ortalama hataları (DSAS sonuçları)

YÖNTEM	DSAS Sonuçları	
	metre	piksel
SOM	14,61	0,49
ANN	18,31	0,61
SOM – ANN	6,69	0,22
RANDOM FOREST	13,64	0,45
SOM – RANDOM FOREST	23,69	0,79

Çizelge 4. 22' de yer alan karşılaştırma sonuçları incelendiğinde, en iyi sonuç SOM – ANN yöntemiyle, en kötü sonuç ise SOM – RANDOM FOREST yöntemi ile elde edilmiştir. Çizelge 4. 22' de yer alan değerler, SOM yöntemi sonuçları kullanılarak rastgele toplanan eğitim verileri ile eğitilen ağların (SOM – ANN), kullanıcı tarafından toplanan eğitim verileri ile eğitilen ağlara (ANN) göre daha iyi sonuçlar ortaya koyduğunu göstermektedir.

#### 4.4.4 Üçüncü Aşama

Üçüncü aşamada, birinci ve ikinci aşamada tasarlanan ağ kombinasyonlarından en iyi sonucu veren TRAINSCG – TANSIG kombinasyonu kullanılarak farklı sayıda gizli katman ve iterasyon sayılarıyla yeni ağlar tasarlanmıştır. Bu ağların eğitimi için birinci aşamadaki ANN ağlarının ve ikinci aşamadaki SOM – ANN ağlarının eğitim verileri kullanılmıştır. Farklı gizli katman ve iterasyon sayılarından oluşan kombinasyonlar Çizelge 4. 23' de gösterilmiştir.

Birinci aşamadaki ANN ağlarına ait eğitim verilerinin kullanıldığı ağlar ANN-3 ağları, ikinci aşamadaki SOM – ANN ağlarının eğitim verilerinin kullanıldığı ağlar kısaca SOM – ANN-3 ağları olarak isimlendirilmiştir.

Çizelge 4. 23 Gizli katman ve iterasyon sayıları ile oluşturulan kombinasyonlar

Kombinasyon	Gizli Katman Sayısı	İterasyon Sayısı
1	5	500
2	10	500
3	15	500
4	5	1000
5	10	1000
6	15	1000
7	5	1500
8	10	1500
9	15	1500
10	5	2000
11	10	2000
12	15	2000

İterasyon ve gizli katman sayıları ile oluşturulan her bir kombinasyonun ağ yapısı, Sadece Eğitim ve Varsayılan eğitim şekilleriyle tasarlanmıştır.

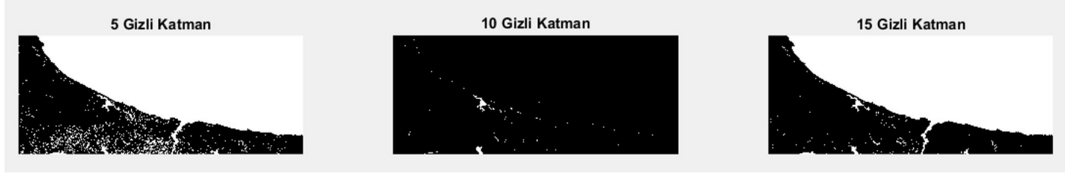
ANN-3 ağlarından, en iyi sonucu veren kombinasyonun belirlenmesi için 6 numaralı görüntü kullanılmıştır.



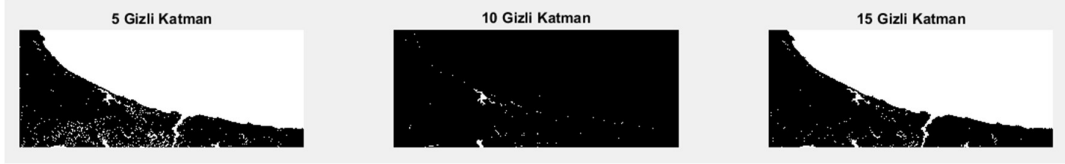
Şekil 4. 42 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan)



Şekil 4. 43 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan)

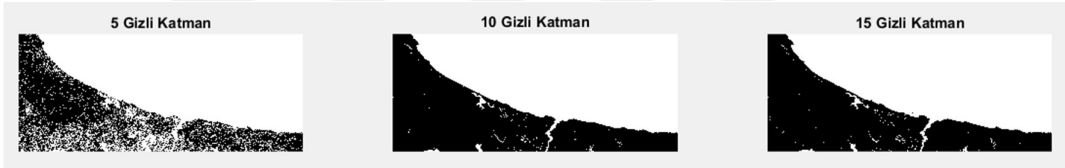


Şekil 4. 44 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan)

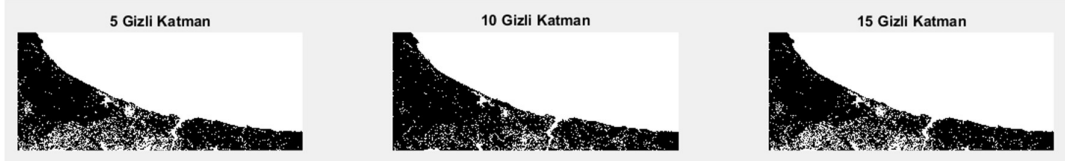


Şekil 4. 45 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Varsayılan)

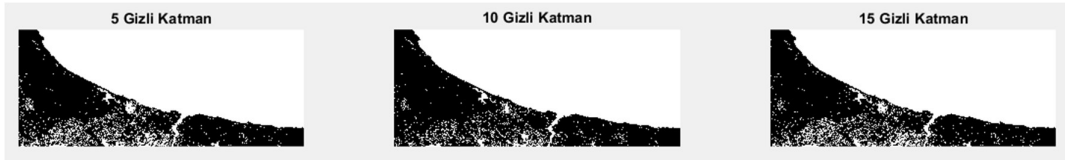
Şekil 4. 42 – Şekil 4. 45 arası şekiller, Varsayılan eğitim şeklini kullanılan ağlara ait sonuçları göstermektedir.



Şekil 4. 46 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim)



Şekil 4. 47 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim)



Şekil 4. 48 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim)



Şekil 4. 49 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (ANN-3 – Sadece Eğitim)

Şekil 4. 46 – Şekil 4. 49 arası şekiller, Sadece Eğitim şeklini kullanılan ağlara ait sonuçları göstermektedir.

ANN-3 ağlarının eğitime süreleri Çizelge 4. 24’ de gösterilmiştir.

Çizelge 4. 24 ANN-3 ağlarının eğitime süreleri

İterasyon – Gizli Katman	Sadece Eğitim	Varsayılan
500 – 5	46 dakika	18 dakika
500 – 10	1 saat 26 dakika	28 dakika
500 – 15	55 dakika	20 dakika
1000 – 5	1 saat 31 dakika	17 dakika
1000 – 10	2 saat 58 dakika	38 dakika
1000 – 15	4 saat 36 dakika	27 dakika
1500 – 5	3 saat 6 dakika	32 dakika
1500 – 10	4 saat 58 dakika	29 dakika
1500 – 15	7 saat 40 dakika	49 dakika
2000 – 5	4 saat 22 dakika	18 dakika
2000 – 10	5 saat 51 dakika	43 dakika
2000 – 15	11 saat 20 dakika	29 dakika

Çizelge 4. 24’ de gösterilen kombinasyonların eğitime süreleri incelendiğinde, en hızlı eğitim 1000 iterasyon-5 gizli katman (Varsayılan) kombinasyonunda ve en yavaş eğitim ise 2000 iterasyon-15 gizli katman (Sadece Eğitim) kombinasyonunda gerçekleşmiştir.

ANN-3 ağlarında kullanılan eğitim verilerinin ağ çıktıları ile hedef çıktı verileri karşılaştırılarak ortalama karesel hatalar hesaplanmış ve Çizelge 4. 25’ de gösterilmiştir.

Çizelge 4. 25 ANN-3 ağlarının ortalama karesel hataları

İterasyon – Gizli Katman	Sadece Eğitim	Varsayılan
500 – 5	2,51e-04	3,50e-04
500 – 10	5,77e-04	8,92e-04
500 – 15	5,15e-04	1,70e-03
1000 – 5	2,43e-04	2,72e-04
1000 – 10	2,44e-04	8,66e-04
1000 – 15	2,54e-04	1,66e-03
1500 – 5	2,39e-04	3,02e-04
1500 – 10	2,41e-04	8,76e-04
1500 – 15	2,49e-04	1,63e-03
2000 – 5	2,39e-04	3,30e-04
2000 – 10	2,40e-04	7,74e-04
2000 – 15	2,50e-04	1,63e-03

Çizelge 4. 25' de verilen ortalama karesel hatalar incelendiğinde, en yüksek ortalama karesel hataya 500 iterasyon-5 gizli katman (Varsayılan) kombinasyonu sahiptir. En düşük karesel ortalama hataya ise 1500 iterasyon-5 gizli katman (Varsayılan) ve 2000 iterasyon-5 gizli katman (Varsayılan) kombinasyonları sahiptir. Ayrıca iki eğitim şeklinde de aynı gizli katman sayısına sahip kombinasyonların yaklaşık aynı ortalama karesel hataya sahip olduğu gözlemlenmiştir.

ANN-3 ağlarında kullanılan eğitim verilerine ait ağ çıktılarının gradyan değerleri Çizelge 4. 26' de sunulmuştur.

Çizelge 4. 26 ANN-3 ağlarının gradyan değerleri

İterasyon – Gizli Katman	Sadece Eğitim	Varsayılan
500 – 5	1,28e-04	1,42e-04
500 – 10	6,67e-03	2,04e-03
500 – 15	5,98e-03	4,74e-03
1000 – 5	1,04e-04	1,82e-04
1000 – 10	4,55e-05	2,78e-04
1000 – 15	3,66e-04	1,52e-04
1500 – 5	8,59e-06	2,78e-04
1500 – 10	9,93e-06	3,11e-03
1500 – 15	9,07e-04	7,63e-04
2000 – 5	6,83e-06	2,97e-04
2000 – 10	1,24e-04	1,71e-03
2000 – 15	9,01e-05	9,35e-04

Çizelge 4. 26' da yer alan gradyan değerlerine bakıldığında, en düşük değer 2000 iterasyon-5 gizli katman (Sadece Eğitim) kombinasyonu ile elde edilirken, en büyük değer ise 500 iterasyon-10 gizli katman (Sadece Eğitim) kombinasyonu ile elde edilmiştir. Gradyan değerinin küçük olması, ağların eğitimlerini tamamlaması için bir kriterdir.

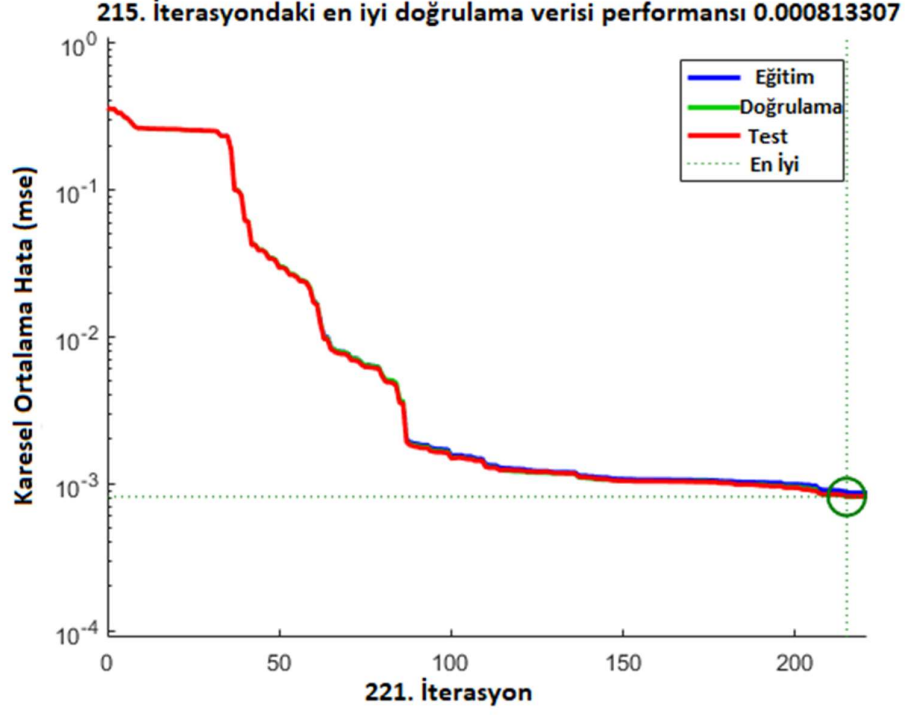
Farklı iterasyon ve gizli katman sayıları ile oluşturulan ağlardan toplam 24 adet görüntü elde edilmiştir. Bu görüntüler incelendiğinde kara ile deniz sınıfının birbirinden ayrılabilirdiği yalnızca 19 görüntü vardır ve yalnızca bunlardan kıyı çizgisi çıkarılabilmektedir. Ardından elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Geriye kalan 5 görüntüden kara ve deniz sınıfları net bir şekilde ayrılamadığı için karşılaştırma yapılamamıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Hesaplanan ortalama hata değerleri Çizelge 4. 27' de gösterilmiştir.

Çizelge 4. 27 ANN-3 ağlarına ait karşılaştırma sonuçlarının ortalama hataları

İterasyon – Gizli Katman	Ortalama Hata (Sadece Eğitim)		Ortalama Hata (Varsayılan)	
	metre	piksel	metre	piksel
500 – 5	-	-	20,58	0,69
500 – 10	14,46	0,48	-	-
500 – 15	14,91	0,50	15,16	0,51
1000 – 5	59,97	2,00	-	-
1000 – 10	32,04	1,07	10,80	0,36
1000 – 15	47,65	1,59	13,99	0,47
1500 – 5	54,21	1,81	24,58	0,82
1500 – 10	41,21	1,37	-	-
1500 – 15	42,61	1,42	14,76	0,49
2000 – 5	52,74	1,76	20,27	0,68
2000 – 10	38,90	1,30	-	-
2000 – 15	40,42	1,35	15,00	0,50

Çizelge 4. 27' deki sonuçlar, ANN-3 ağlarına ait karşılaştırma sonuçlarını göstermektedir. Çizelge 4. 27' de yer alan metre/piksel bazlı ortalama hata değerleri incelendiğinde, en iyi sonuç 1000 iterasyon-10 gizli katman (Varsayılan) kombinasyonu ile ve en kötü sonuç ise 1000 iterasyon-5 gizli katman (Sadece Eğitim) kombinasyonu ile elde edilmiştir.

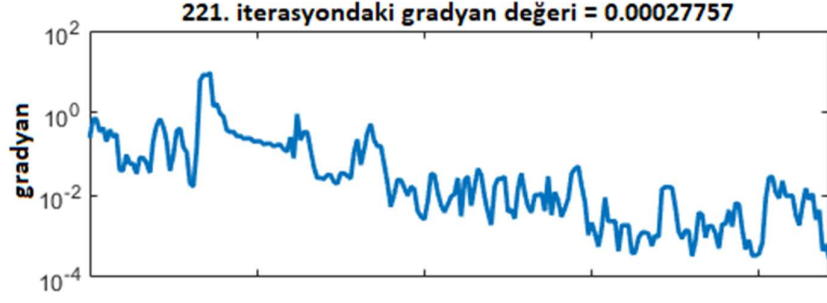
Çizelge 4. 27' de yer alan metre/piksel bazlı ortalama hata değerlerine göre, en iyi sonuç 1000 iterasyon ve 10 gizli katman sayılı Varsayılan eğitim şekli kullanılarak tasarlanan ağdan elde edilmiştir. Bu kombinasyona ait performans grafiği Şekil 4. 50' de sunulmuştur.



Şekil 4. 50 1000 iterasyon - 10 gizli katman (Varsayılan) kombinasyonu performans grafiği

Şekil 4. 50' deki grafikte dikey sütun ortalama karesel hatayı, yatay sütun ise iterasyon sayısını temsil etmektedir. Mavi çizgi eğitim verisine ait ortalama karesel hatayı, yeşil çizgi doğrulama verisine ilişkin ortalama karesel hatayı ve kırmızı çizgi ise test verisinin ortalama karesel hatasını temsil etmektedir. Şekil 4. 50' de eğitim, doğrulama ve test verilerinin karesel ortalama hatalarında ilk başta çok küçük bir değişim görülmektedir. Daha sonra karesel ortalama hata değerlerinde ani bir azalma görülmektedir. Son olarak da yavaş yavaş azalarak optimum değere ulaşmıştır. Tasarlanan bu ağ 215. iterasyonda doğrulama verisinin en iyi performansına göre eğitimini tamamlamış ve 2000 iterasyonun tamamını yapmamıştır. 215. iterasyondaki performansını 221. iterasyona kadar 6 defa test etmiştir ve değişim olmadığı için eğitimi durdurmuştur. Ayrıca bu ağın iterasyon sayısına bağlı olarak gradyan değerlerindeki değişimi gösteren grafik Şekil 4. 51' de sunulmuştur.





Şekil 4. 51 1000 iterasyon - 10 gizli katman (Varsayılan) kombinasyonu gradyan değerleri grafiği

Şekil 4. 50' deki ağa ait her iterasyondaki gradyan değerlerinin yer aldığı grafik Şekil 4. 51'de sunulmuştur. Gradyan değerleri, her iterasyonda ağdan çıkan tahmini değerlerin gradyan değerleridir ve belirlenen gradyan değerine doğru azalarak devam eder.

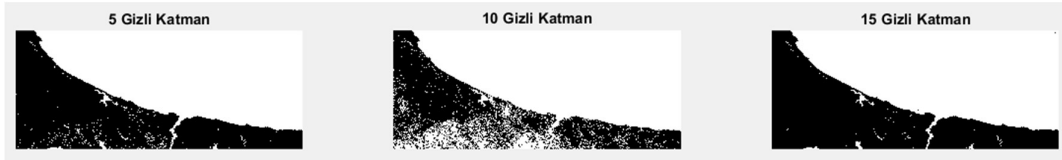
SOM – ANN-3 ağları da Çizelge 4. 23' deki kombinasyonlar kullanılarak oluşturulmuştur. SOM – ANN-3 ağlarından, en iyi sonucu veren kombinasyonun belirlenmesi için 6 numaralı görüntü kullanılmıştır.



Şekil 4. 52 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan)



Şekil 4. 53 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan)

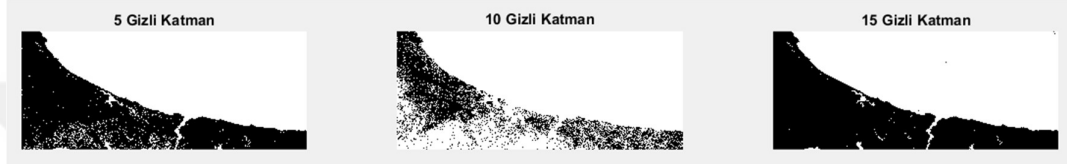


Şekil 4. 54 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan)

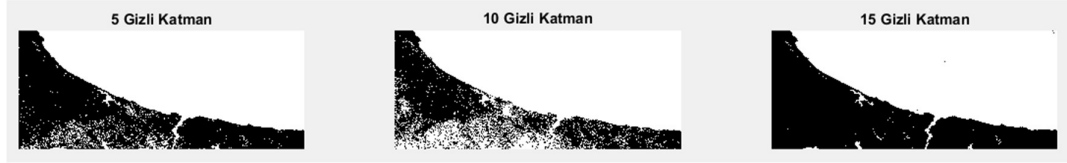


Şekil 4. 55 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı SOM – ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Varsayılan)

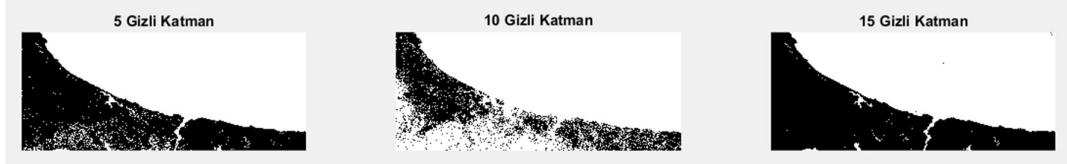
Şekil 4. 52 – Şekil 4. 55 arası şekiller, Varsayılan eğitim şeklini kullanan ağlara ait sonuçları göstermektedir.



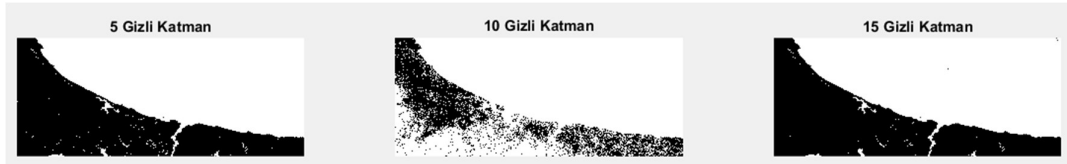
Şekil 4. 56 500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim)



Şekil 4. 57 1000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim)



Şekil 4. 58 1500 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim)



Şekil 4. 59 2000 iterasyon ve sırasıyla 5-10-15 gizli katman sayılı ANN ağlarının sonuç görüntüleri (SOM – ANN-3 – Sadece Eğitim)

Şekil 4. 56 – Şekil 4. 59 arası şekiller, Sadece Eğitim şeklini kullanılan ağlara ait sonuçları göstermektedir.

SOM – ANN-3 ağlarının eğitilme süreleri Çizelge 4. 28’ de gösterilmiştir.

Çizelge 4. 28 SOM – ANN-3 ağlarının eğitilme süreleri

İterasyon – Gizli Katman	Sadece Eğitim	Varsayılan
500 – 5	48 dakika	29 dakika
500 – 10	1 saat 32 dakika	13 dakika
500 – 15	2 saat 14 dakika	43 dakika
1000 – 5	1 saat 35 dakika	7 dakika
1000 – 10	3 saat 15 dakika	33 dakika
1000 – 15	1 saat 51 dakika	21 dakika
1500 – 5	2 saat 17 dakika	7 dakika
1500 – 10	4 saat 29 dakika	19 dakika
1500 – 15	2 saat 43 dakika	25 dakika
2000 – 5	3 saat 7 dakika	5 dakika
2000 – 10	6 saat 30 dakika	42 dakika
2000 – 15	8 saat 36 dakika	42 dakika

Çizelge 4. 28' de gösterilen kombinasyonların eğitilme süreleri incelendiğinde, en hızlı eğitim 2000 iterasyon-5 gizli katman (Varsayılan) kombinasyonunda ve en yavaş eğitim ise 2000 iterasyon-15 gizli katman (Sadece Eğitim) kombinasyonunda gerçekleşmiştir.

SOM – ANN-3 ağlarında kullanılan eğitim verilerinin ağ çıktıları ile hedef çıktı verileri karşılaştırılarak ortalama karesel hatalar hesaplanmış ve Çizelge 4. 29' da gösterilmiştir.

Çizelge 4. 29 SOM – ANN-3 ağlarının ortalama karesel hataları

İterasyon – Gizli Katman	Sadece Eğitim	Varsayılan
500 – 5	1,73e-03	1,79e-03
500 – 10	3,30e-03	4,17e-03
500 – 15	3,43e-03	3,50e-03
1000 – 5	1,61e-03	3,39e-03
1000 – 10	1,81e-03	3,48e-03
1000 – 15	3,41e-03	3,54e-03
1500 – 5	1,55e-03	3,40e-03
1500 – 10	1,52e-03	3,47e-03
1500 – 15	3,40e-03	3,52e-03
2000 – 5	1,51e-03	3,50e-03
2000 – 10	1,48e-03	3,13e-03
2000 – 15	3,41e-03	3,54e-03

Çizelge 4. 29' da verilen ortalama karesel hatalar incelendiğinde, en yüksek ortalama karesel hataya 500 iterasyon-10 gizli katman (Varsayılan) kombinasyonu sahiptir. En düşük karesel ortalama hataya ise 1500 iterasyon-5 gizli katman (Varsayılan) ve 2000 iterasyon-10 gizli katman (Sadece Eğitim) kombinasyonları sahiptir. Ayrıca iki eğitim şeklinde de aynı gizli katman sayısına sahip kombinasyonların yaklaşık aynı ortalama karesel hataya sahip olduğu gözlemlenmiştir.

SOM – ANN-3 ağlarında kullanılan eğitim verilerine ait ağ çıktılarının gradyan değerleri Çizelge 4. 30' da sunulmuştur.

Çizelge 4. 30 SOM – ANN-3 ağlarının gradyan değerleri

İterasyon – Gizli Katman	Sadece Eğitim	Varsayılan
500 – 5	8,92e-04	4,19e-03
500 – 10	2,56e-03	1,53e-03
500 – 15	5,74e-04	3,08e-03
1000 – 5	1,81e-03	1,56e-03
1000 – 10	4,53e-03	2,05e-02
1000 – 15	6,44e-03	1,51e-03
1500 – 5	9,55e-04	2,82e-04
1500 – 10	1,55e-03	1,33e-03
1500 – 15	3,80e-04	1,76e-03
2000 – 5	7,68e-04	4,71e-04
2000 – 10	2,77e-04	7,93e-04
2000 – 15	9,65e-04	6,43e-04

Çizelge 4. 30' da yer alan gradyan değerlerine bakıldığında, en düşük değer 2000 iterasyon-10 gizli katman (Sadece Eğitim) kombinasyonu ile elde edilirken, en büyük değer ise 1000 iterasyon-10 gizli katman (Varsayılan) kombinasyonu ile elde edilmiştir. Gradyan değerinin küçük olması, ağların eğitimlerini tamamlaması için bir kriterdir.

Farklı iterasyon ve gizli katman sayıları ile oluşturulan ağlardan toplam 24 adet görüntü elde edilmiştir. Bu görüntüler incelendiğinde kara ile deniz sınıfının birbirinden ayrılabilirdiği yalnızca 16 görüntü vardır ve yalnızca bunlardan kıyı çizgisi çıkarılabilmektedir. Ardından elle yapılan sayısallaştırma sonuçları ile karşılaştırılmıştır. Geriye kalan 8 görüntüden kara ve deniz sınıfları net bir şekilde ayıramadığı için karşılaştırma yapılamamıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır. Hesaplanan ortalama hata değerleri Çizelge 4. 31' de gösterilmiştir.

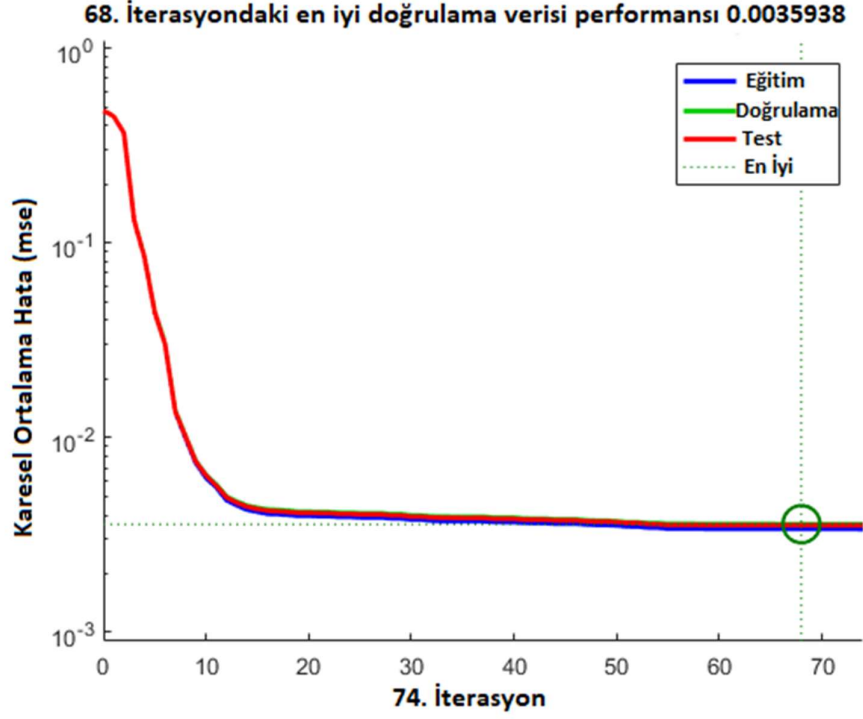
Çizelge 4. 31 SOM – ANN-3 ağlarına ait karşılaştırma sonuçlarının ortalama hataları

İterasyon – Gizli Katman	Ortalama Hata (Sadece Eğitim)		Ortalama Hata (Varsayılan)	
	metre	piksel	metre	piksel
500 – 5	24,69	0,82	10,54	0,35
500 – 10	-	-	-	-
500 – 15	9,00	0,30	9,05	0,30
1000 – 5	29,13	0,97	5,91	0,20
1000 – 10	-	-	-	-
1000 – 15	7,18	0,24	12,87	0,43
1500 – 5	24,99	0,83	18,28	0,61
1500 – 10	-	-	-	-
1500 – 15	7,09	0,24	6,27	0,21
2000 – 5	11,88	0,40	13,79	0,46
2000 – 10	-	-	-	-
2000 – 15	6,51	0,22	5,93	0,20

Çizelge 4. 31' deki sonuçlar, SOM – ANN-3 ağlarına ait karşılaştırma sonuçlarını göstermektedir. Çizelge 4. 31' de yer alan metre/piksel bazlı ortalama hata değerleri incelendiğinde, en iyi sonuç 1000 iterasyon-5 gizli katman (Varsayılan) kombinasyonu ve en kötü sonuç ise 1000 iterasyon-5 gizli katman (Sadece Eğitim) kombinasyonu ile elde edilmiştir.

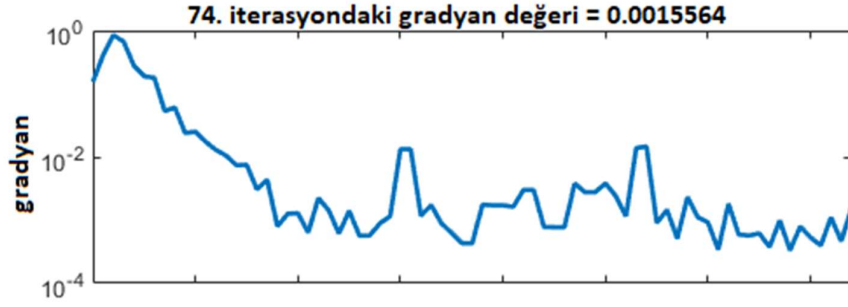
Çizelge 4. 31' deki sonuçlar, SOM – ANN-3 ağlarına ait karşılaştırma sonuçlarını göstermektedir.

Çizelge 4. 31' de yer alan metre/piksel bazlı ortalama hat değerlerine göre, en iyi sonuç 1000 iterasyon ve 5 gizli katman sayılı Varsayılan eğitim şekli kullanılarak tasarlanan ağdan elde edilmiştir. Bu kombinasyona ait performans grafiği Şekil 4. 60' da sunulmuştur.



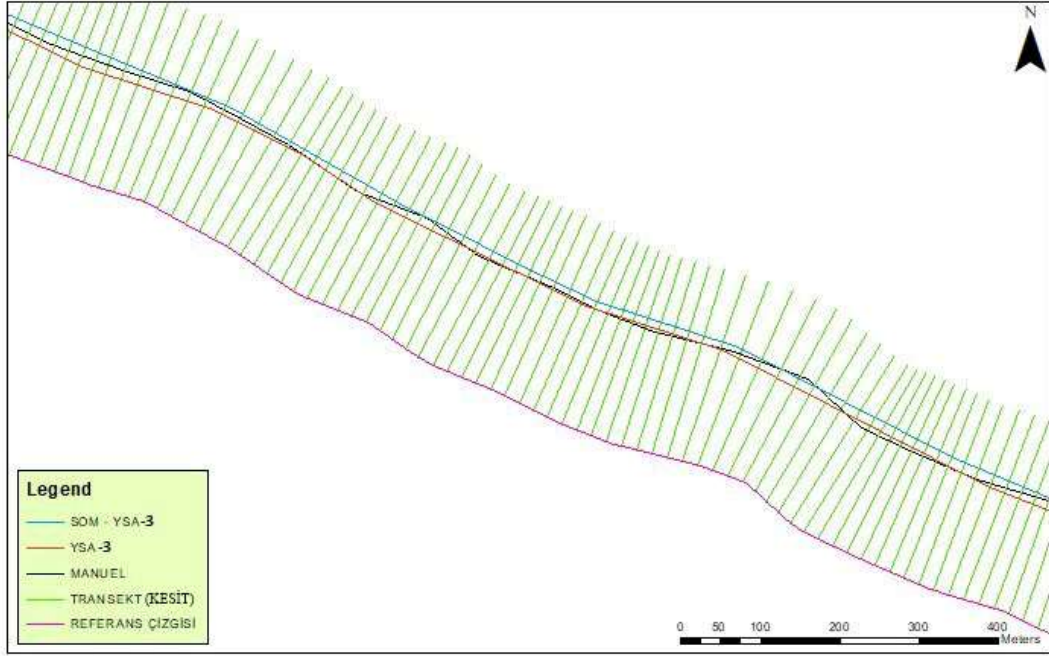
Şekil 4. 60 1000 iterasyon - 5 gizli katman (Varsayılan) kombinasyonu performans grafiği

Şekil 4. 60' daki grafikte dikey sütun ortalama karesel hatayı, yatay sütun ise iterasyon sayısını temsil etmektedir. Mavi çizgi eğitim verisine ait ortalama karesel hatayı, yeşil çizgi doğrulama verisine ilişkin ortalama karesel hatayı ve kırmızı çizgi ise test verisinin ortalama karesel hatasını temsil etmektedir. Şekil 4. 60' da eğitim, doğrulama ve test verilerinin karesel ortalama hatalarında ilk başta hızlı bir azalma görülmektedir. Daha sonra ise, karesel ortalama hata değerlerinde çok büyük bir değişim görülmemektedir ve optimum sonuca bu şekilde ulaşmıştır. Tasarlanan bu ağ 68. iterasyonda doğrulama verisinin en iyi performansına göre eğitimini tamamlamış ve 2000 iterasyonun tamamını yapmamıştır. 68. iterasyondaki performansını 74. iterasyona kadar 6 defa test etmiştir ve değişim olmadığı için eğitimi durdurmuştur. Ayrıca bu ağın iterasyon sayısına bağlı olarak gradyan değerlerindeki değişimi gösteren grafik Şekil 4. 61' de sunulmuştur.



Şekil 4. 61 1000 iterasyon - 5 gizli katman (Varsayılan) kombinasyonu gradyan değerleri grafiği

Şekil 4. 60' daki ağa ait her iterasyondaki gradyan değerlerinin yer aldığı grafik Şekil 4. 61'de sunulmuştur. Gradyan değerleri, her iterasyonda ağdan çıkan tahmini değerlerin gradyan değerleridir ve belirlenen gradyan değerine doğru azalarak devam eder.



Şekil 4. 62 Üçüncü aşamada en iyi sonucu veren kıyı çizgileri ve DSAS bileşenleri

Şekil 4. 62' de üçüncü aşamada yapılan karşılaştırma sonuçlarına göre en iyi sonuç veren kıyı çizgileri, elle yapılan sayısallaştırma, referans kıyı çizgisi ve kesitler yer almaktadır.

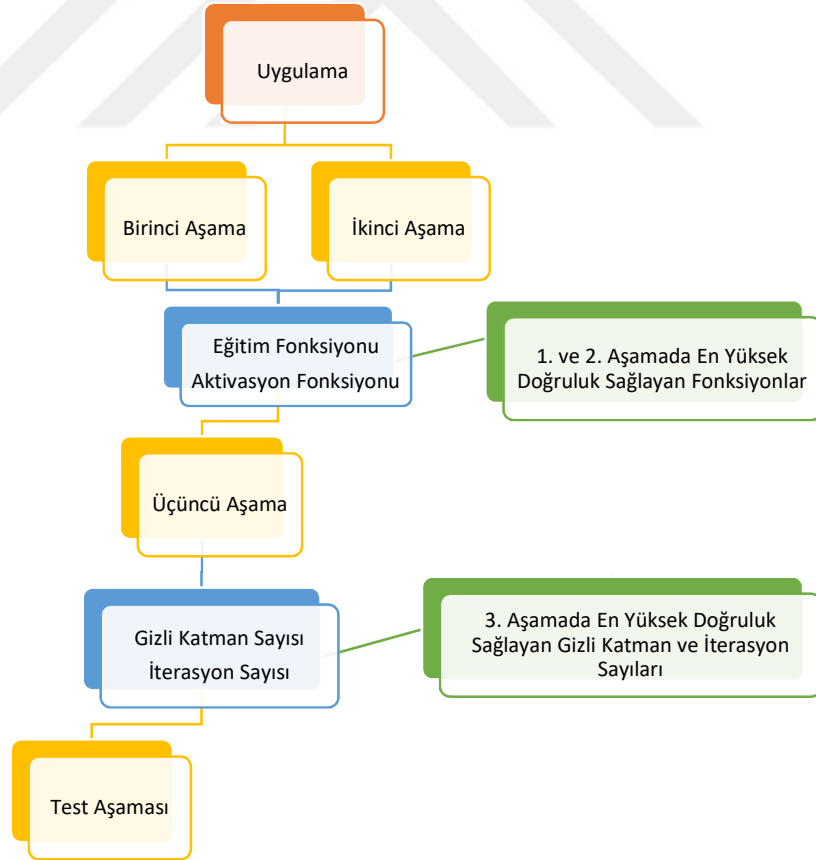
Şekil 4. 62' de yer alan kıyı çizgilerine ait ortalama hata değerleri de Çizelge 4. 32' de gösterilmiştir.

Çizelge 4. 32 Üçüncü aşamada en iyi sonuç veren kıyı çizgilerinin ortalama hataları (DSAS sonuçları)

YÖNTEM	DSAS Sonuçları	
	metre	piksel
ANN-3	10,80	0,36
SOM – ANN-3	5,91	0,20

Çizelge 4. 32' de yer alan değerler, SOM yöntemi sonuçları kullanılarak rastgele toplanan eğitim verileri ile eğitilen ağların (SOM – ANN-3), kullanıcı tarafından toplanan eğitim verileri ile eğitilen ağlara (ANN-3) göre daha iyi sonuçlar ortaya koyduğunu göstermektedir.

#### 4.4.5 Test Aşaması



Şekil 4. 63 Test aşamasında kullanılan ANN bileşenlerin elde edildiği aşamaların gösterimi



Birinci, ikinci ve üçüncü aşamalarda tasarlanan ağ kombinasyonlarının metre/piksel bazlı ortalama hata değerleri incelendiğinde, en iyi ağ kombinasyonunun üçüncü aşamadaki 5 gizli katmanlı, 1000 iterasyonlu SOM – ANN-3 ağı olduğu gözlemlenmiştir. Bu ağ yapısının özelliklerinin hangi aşamalarda belirlendiği Şekil 4. 63' te gösterilmiştir.

En iyi sonucu veren bu ağ yapısı 7, 8, 9, 10, 11 ve 12 numaralı görüntüler kullanılarak test edilmiştir. Test sonuçlarından elde edilen kıyılar, elle yapılan sayısallaştırma sonuçlarıyla karşılaştırılmış ve ortalama hatalar hesaplanmıştır. Karşılaştırma için Sayısal Kıyı Çizgisi Analizi (DSAS) kullanılmıştır. DSAS için gerekli olan referans çizgisi, elle yapılan sayısallaştırmaya 150 m tampon (buffer) uygulanarak elde edilmiştir. Kesitlerin boyutları 250 m ve iki kesit arası mesafe 5 m olacak şekilde ayarlanmıştır.



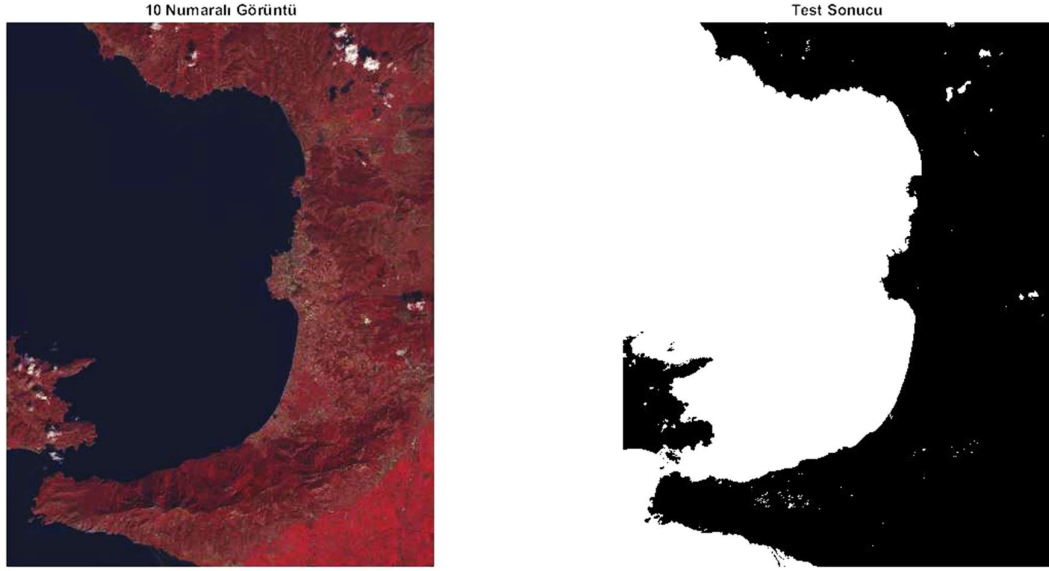
Şekil 4. 64 7 numaralı test görüntüsü ve elde edilen sonucu



Şekil 4. 65 8 numaralı test görüntüsü ve elde edilen sonucu



Şekil 4. 66 9 numaralı test görüntüsü ve elde edilen sonucu



Şekil 4. 67 10 numaralı test görüntüsü ve elde edilen sonucu

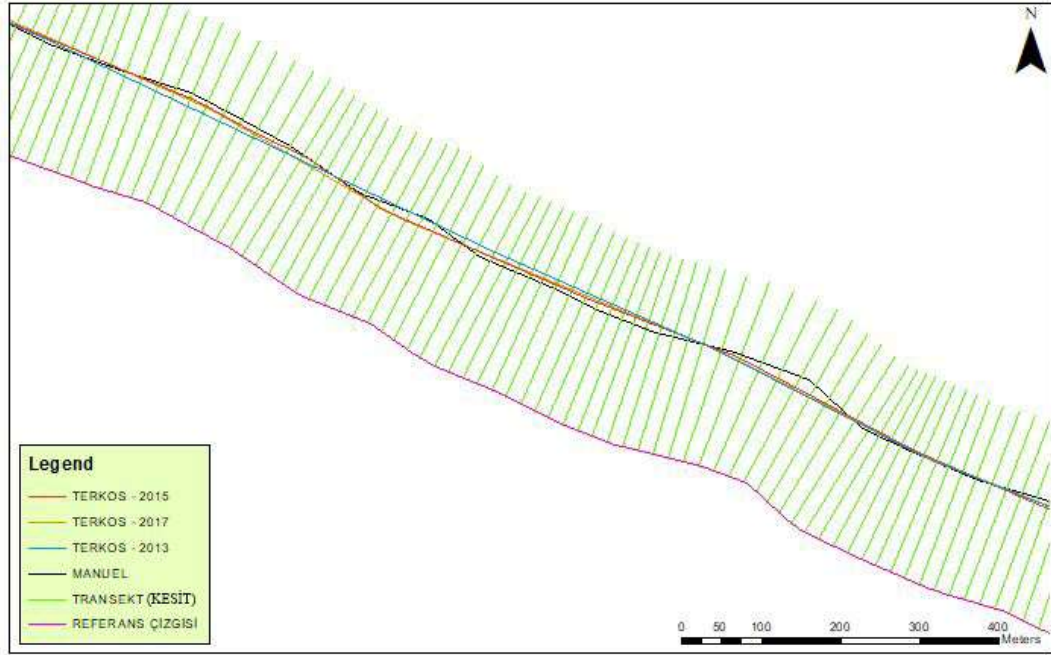


Şekil 4. 68 11 numaralı test görüntüsü ve elde edilen sonucu



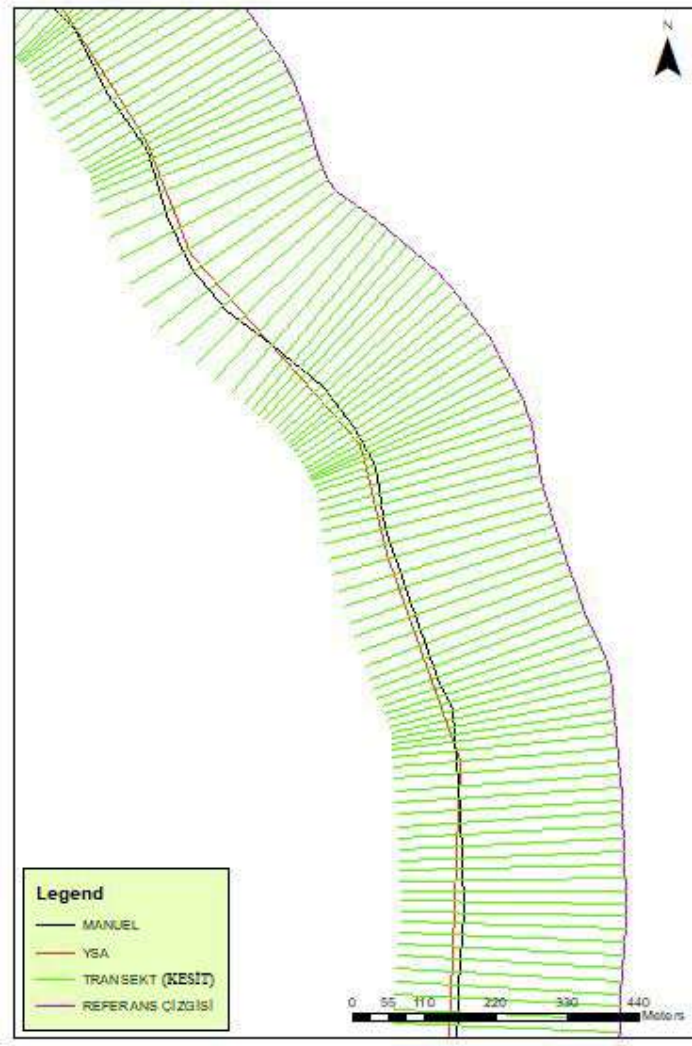
Şekil 4. 69 12 numaralı test görüntüsü ve elde edilen sonucu

Şekil 4. 64 ve Şekil 4. 69 arası şekillerde, sırasıyla 7, 8, 9, 10, 11 ve 12 numaralı görüntüler ve elde edilen sınıflandırılmış sonuç görüntüler yer almaktadır.

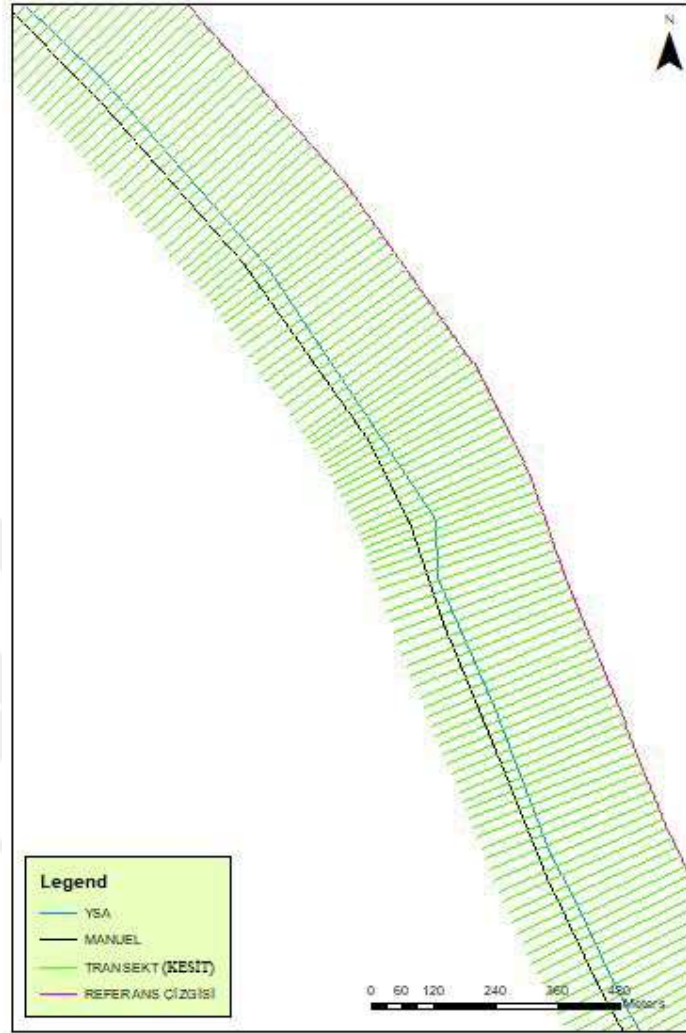


Şekil 4. 70 Test aşamasında kullanılan Terkos bölgesine ait kıyı çizgileri ve DSAS bileşenleri

Test aşamasında kullanılan Terkos bölgesine ait 2017, 2015 ve 2013 tarihli görüntülerden edilen kıyı çizgileri, elle yapılan sayısallaştırma, referans çizgisi ve kesitler Şekil 4. 70' de gösterilmiştir.

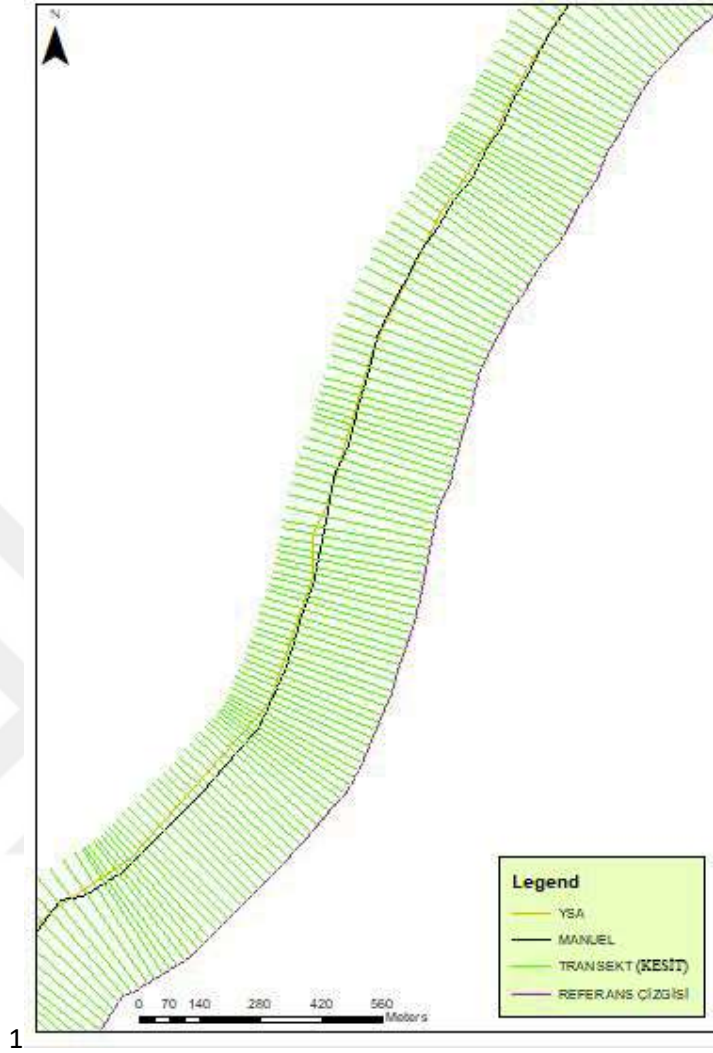


Şekil 4. 71 Test aşamasında kullanılan İzmir bölgesine ait kıyı çizgisi ve DSAS bileşenleri  
Şekil 4. 71' de test aşamasında kullanılan İzmir bölgesine ait görüntüden elde edilen kıyı çizgisi, elle yapılan sayısallaştırma, referans çizgisi ve kesitler gösterilmiştir.



Şekil 4. 72 Test aşamasında kullanılan Antalya bölgesine ait kıyı çizgisi ve DSAS bileşenleri

Şekil 4. 72' de test aşamasında kullanılan Antalya bölgesine ait görüntüden elde edilen kıyı çizgisi, elle yapılan sayısallaştırma, referans çizgisi ve kesitler gösterilmiştir.



1

Şekil 4. 73 Test aşamasında kullanılan Erçek Gölü' ne ait kıyı çizgisi ve DSAS bileşenleri  
Şekil 4. 73' de test aşamasında kullanılan Erçek Gölü' ne ait görüntüden elde edilen kıyı çizgisi, elle yapılan sayısallaştırma, referans çizgisi ve kesitler gösterilmiştir.

Son olarak, test aşamasına ait Şekil 4. 70 ve Şekil 4. 73 arası şekillerde bulunan kıyı çizgilerinin ortalama hata değerleri de Çizelge 4. 33' de gösterilmiştir.

Çizelge 4. 33 Test aşamasına ait kıyı çizgilerinin ortalama hataları (DSAS sonuçları)

Görüntü İsmi	Görüntü Numarası	DSAS Sonuçları	
		metre	piksel
Terkos 2017	7	10,90	0,36
Terkos 2015	8	10,80	0,36
Terkos 2013	9	13,98	0,47
İzmir 2017	10	9,26	0,31
Antalya 2017	11	39,31	1,31
Erçek Gölü 2017	12	16,99	0,57

Erçek Gölü' nün elle yapılan sayısallaştırma sonucu hesaplanan alanı 109,61 km<sup>2</sup>, test sonucundan ise gölün alanı 108,69 km<sup>2</sup> olarak hesaplanmıştır. İki alan değeri arasındaki fark 0,92 km<sup>2</sup> olarak hesaplanmıştır.

Çizelge 4. 33' de yer alan karşılaştırma sonuçlarına bakıldığında, tez kapsamında tespit edilen en iyi sonucu veren ağ yapısının oldukça başarılı sonuçlar ortaya koyduğu görülmektedir. Test görüntülerinin karşılaştırma sonuçlarına göre en iyi sonuç İzmir bölgesine ait görüntüden elde edilmiştir. En kötü sonuç ise Antalya bölgesine ait görüntüden elde edilmiştir.

### SONUÇ VE ÖNERİLER

Sunulan tez kapsamında, birkaç soruya cevap bulmaya yönelik çalışmalar yapılmıştır.

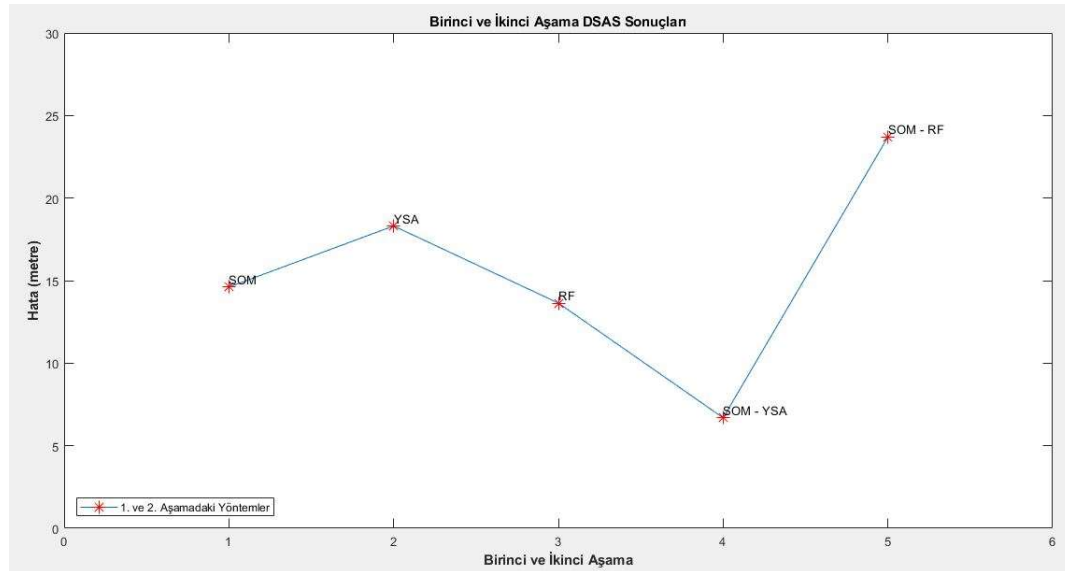
Bu sorulardan birincisi SOM yönteminin kıyı çizgisi çıkarımında kullanılmasının uygun olup olmadığı sorusudur. SOM yönteminin uydu görüntülerine uygulanması sonucunda, 14,61 m (0,49 piksel) doğruluğa sahip kıyı çizgileri elde edilmiştir. Bu doğruluk değeri, kullanılan LANDSAT 8 uydu görüntüsünün çözünürlüğünün yaklaşık yarısına eşittir. Bu bilgiler ışığında, SOM yönteminin tek başına kıyı çizgisi çıkarımında kullanılmasının uygun olduğu gözlemlenmiştir.

Cevabını bulmaya yönelik çalışmalar yapılan ikinci soru, farklı uydu görüntülerinden toplanan eğitim verileri kullanılarak tasarlanan yapay sinir ağları ve ağaç yapılarının hiç kullanılmayan görüntülerde nasıl sonuçlar elde edildiği sorusudur. ANN ve RF yöntemlerinin uydu görüntülerine uygulanması sonucunda, ANN ağlarından 18,31 m (0,61 piksel) ve RF ağaç yapılarından 13,64 m (0,45 piksel) doğruluğa sahip kıyı çizgileri elde edilmiştir. Bu doğruluk değerleri, kullanılan LANDSAT 8 uydu görüntüsünün çözünürlüğünün yaklaşık yarısına eşittir. Bu bilgiler ışığında, ANN ve RF yöntemlerinin eğitimlerinde hiç kullanılmayan görüntülerde iyi sonuçlar verdiği gözlemlenmiştir ve bu yöntemlerin kıyı çizgisi çıkarımında kullanılmasının uygun olduğu sonucuna ulaşılmıştır.



Cevabı aranan üçüncü soru ise, farklı uydu görüntülerinden SOM yöntemi sonuçları kullanılarak toplanan eğitim verileri ile eğitilen yapay sinir ağlarının ve ağaç yapılarının hiç kullanılmayan görüntülerde nasıl sonuçlar elde edildiği sorusudur. SOM – ANN ve SOM – RF yöntemlerinin uydu görüntülerine uygulanması sonucunda, SOM – ANN ağlarından 6,69 m (0,22 piksel), SOM – RF ağaç yapılarından 23,69 m (0,79 piksel) doğruluğa sahip kıyı çizgileri elde edilmiştir. Bu doğruluk sonuçları incelendiğinde, SOM – ANN sonucunun LANDSAT 8 uydu görüntüsünün çözünürlüğünün yaklaşık dörtte birine eşit olduğu görülmektedir. SOM yöntemi sonuçları kullanılarak tasarlanan yapay sinir ağlarının doğruluğu yüksek sonuçlar ortaya koyduğu gözlemlenmiştir. Bu sonuçlar göz önünde bulundurularak, SOM – ANN ağlarının kıyı çizgisi çıkarımında kullanılmasının uygun olduğu sonucuna ulaşılmıştır. SOM – RF yönteminin doğruluk sonucu ise LANDSAT 8 uydu görüntüsünün çözünürlüğünün yaklaşık dörtte üçüne eşit olduğu görülmektedir. Yapılacak çalışmalarda istenilen doğruluk kriterine bağlı olmakla beraber, SOM – RF yöntemi sonuçları kullanılabilir, fakat RF yöntemine göre SOM – RF yönteminden elde edilen sonuçlar daha az doğruluğa sahiptir.

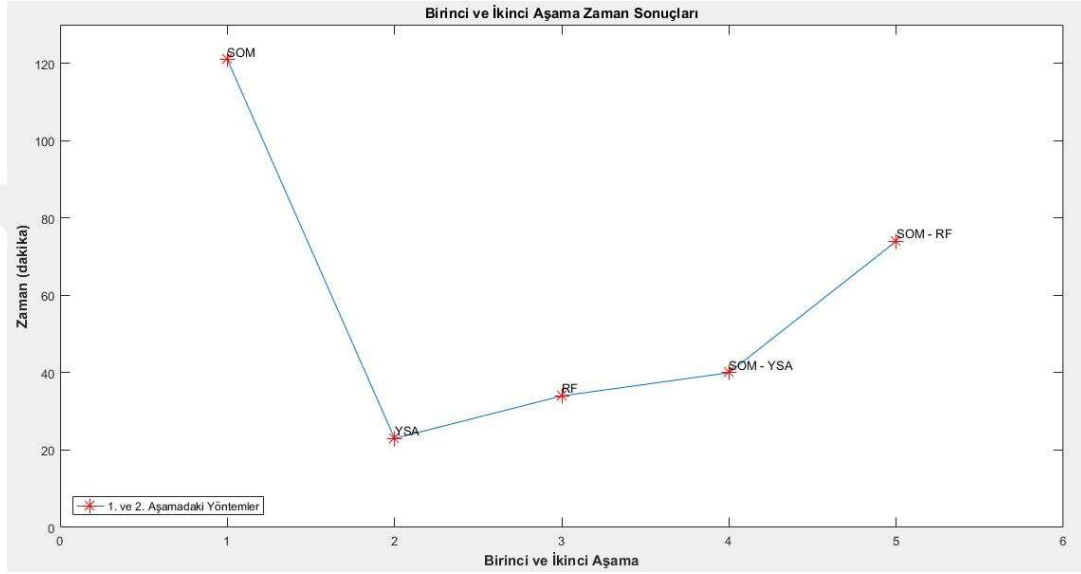
Uygulama bölümünün birinci ve ikinci aşamalarında yer alan SOM, ANN, RF, SOM – ANN ve SOM – RF yöntemlerine ait kombinasyonlardan, en iyi karşılaştırma sonuçlarını sahip kombinasyonların karşılaştırma sonuçlarının grafik üzerindeki dağılımı Şekil 5. 1' de gösterilmiştir.



Şekil 5. 1 Birinci ve ikinci aşamaya ait yöntemlerin en iyi DSAS sonuçları

Şekil 5. 1' deki grafik incelendiğinde, en yüksek ortalama hata değeri SOM – RF yöntemine ait olduğu gözlemlenmektedir. En düşük ortalama hata değeri ise SOM – ANN yöntemi ile elde edilmiştir.

Şekil 5. 1' deki sonuçların elde edilme sürelerinin bulunduğu grafik ise Şekil 5. 2' de sunulmuştur.



Şekil 5. 2 Birinci ve ikinci aşamaya ait en iyi sonuçların elde edilme süreleri

Şekil 5. 2' deki grafiğe göre, SOM yöntemi sonucuna 121 dakikada, ANN ağı sonucuna 23 dakikada, RF ağaç yapısı sonucuna 34 dakikada, SOM – ANN ağı sonucuna 40 dakikada ve SOM – RF ağaç yapısı sonucuna ise 74 dakikada ulaşılmıştır.

Şekil 5. 1 ve Şekil 5. 2' deki grafikler birlikte değerlendirildiğinde, en kısa sürede en iyi sonuca ANN yöntemiyle ulaşılmıştır.

Şekil 5. 1' de karşılaştırma sonuçları yer alan kıyı çizgilerinin kullanıcı tarafından düzeltilerek hatasız hale getirilmesi için harcanan süreler Çizelge 5. 1 'de gösterilmiştir.

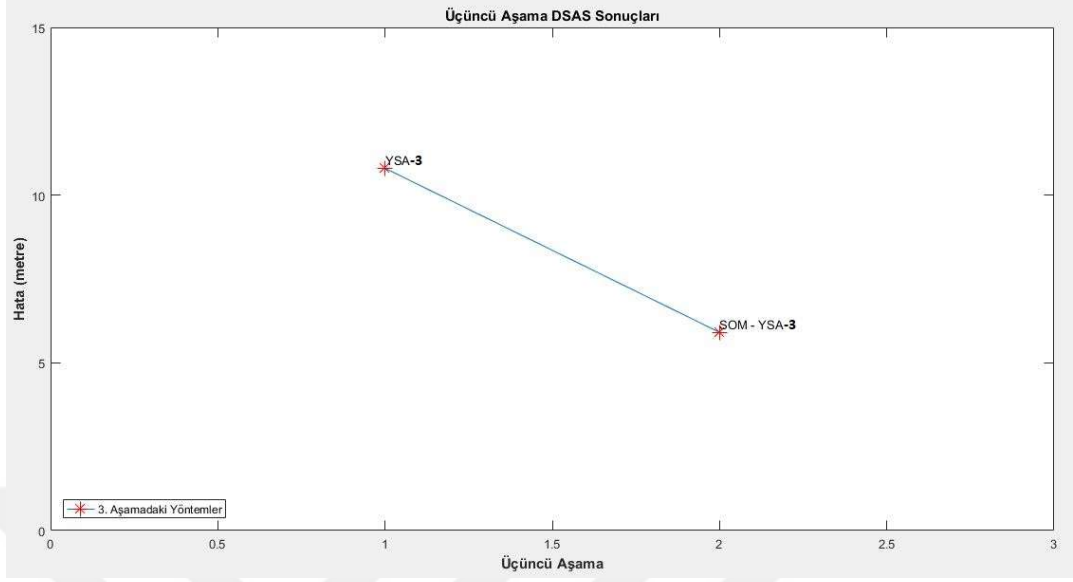
Çizelge 5. 1 Şekil 5. 1' deki kıyı çizgilerinin hatasız hale getirilmesi için harcanan süreler

Yöntem	Düzeltilme Süresi (dakika)
SOM	23 dakika
ANN	31 dakika
RF	21 dakika
SOM – ANN	10 dakika
SOM – RF	41 dakika

Çizelge 5. 1' deki değerler incelendiğinde, uygulanan yöntemlerin karşılaştırma sonuçlarıyla doğru orantılı olarak değiştiği gözlemlenmiştir.

Uygulama bölümünde anlatılan üçüncü aşamada, SOM – ANN yöntemine ait farklı gizli katman iterasyon sayılarının oluşturduğu kombinasyonların sonuçları karşılaştırılmıştır. Bu ağ kombinasyonları hem kullanıcı tarafından toplanan hem de SOM yöntemi sonuçları kullanılarak otomatik toplanan eğitim verileri ile eğitilmiş ve sonuçlar elde edilmiştir. Kullanıcı tarafından toplanan eğitim verilerinin kullanıldığı kombinasyonlardan (ANN-3 ağları) elde edilen en iyi sonuç 10,80 m (0,36 piksel) şeklindedir. SOM yöntemi sonuçları kullanılarak otomatik toplanan eğitim verilerinin kullanıldığı kombinasyonlardan (SOM – ANN-3 ağları) elde edilen en iyi sonuç 5,91 m (0,20 piksel) şeklindedir.

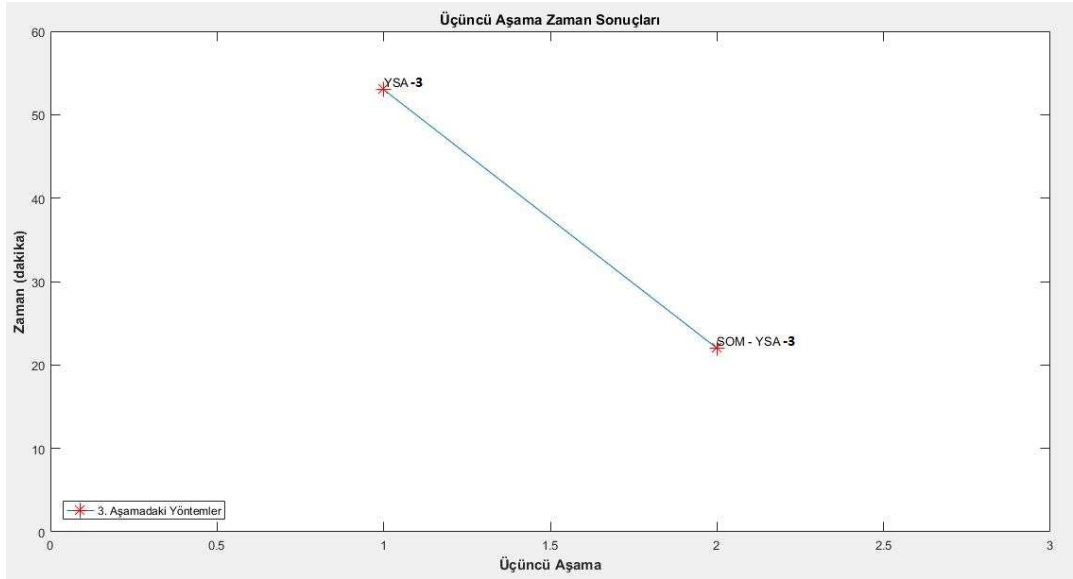
Uygulama bölümünün üçüncü aşamasında yer alan ANN-3 ve SOM – ANN-3 yöntemlerine ait kombinasyonlardan, en iyi karşılaştırma sonuçlarını sahip kombinasyonların karşılaştırma sonuçlarının grafik üzerindeki dağılımı Şekil 5. 3' de gösterilmiştir.



Şekil 5. 3 Üçüncü aşamaya ait en iyi DSAS sonuçları

Şekil 5. 3' deki grafik incelendiğinde, SOM – ANN-3 yönteminin ANN-3 yöntemine göre daha doğru sonuçlar ortaya koyduğu gözlemlenmiştir.

Şekil 5. 3' deki sonuçların elde edilme sürelerinin bulunduğu grafik ise Şekil 5. 4' de sunulmuştur.



Şekil 5. 4 Üçüncü aşamaya ait en iyi sonuçların elde edilme süreleri

Şekil 5. 4' de yer alan grafiğe göre, ANN-3 yöntemi sonucuna 53 dakikada ve SOM – ANN-3 yöntemi sonucuna 22 dakikada ulaşılmıştır.

Şekil 5. 3 ve Şekil 5. 4' deki grafikler birlikte değerlendirildiğinde, en kısa sürede en iyi sonuca SOM – ANN-3 yöntemiyle ulaşılmıştır.

Şekil 5. 3' de karşılaştırma sonuçları yer alan kıyı çizgilerinin kullanıcı tarafından düzeltilerek hatasız hale getirilmesi için harcanan süreler Çizelge 5. 2 'de gösterilmiştir.

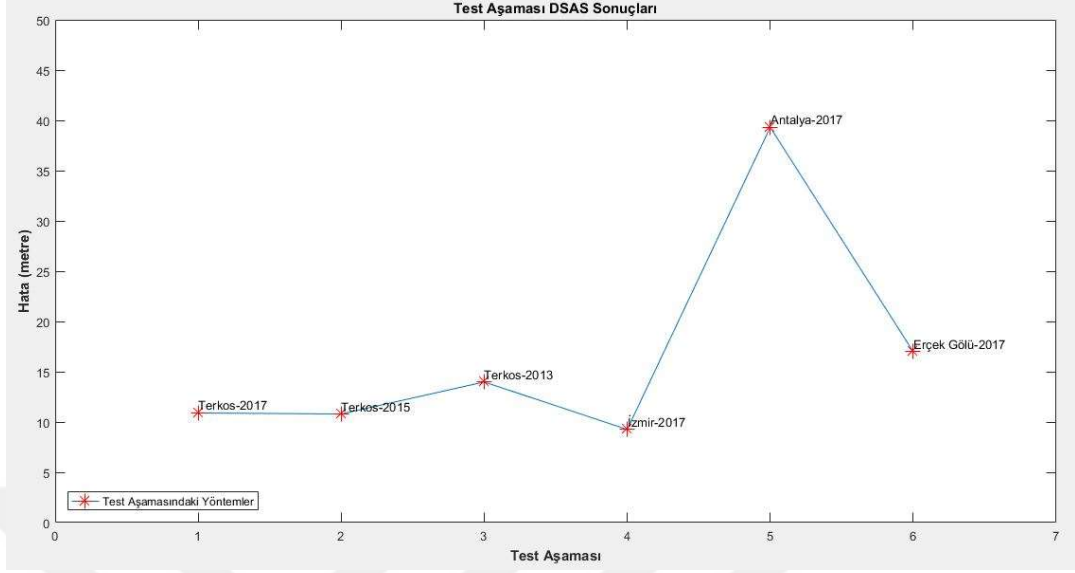
Çizelge 5. 2 Şekil 5. 3' deki kıyı çizgilerinin hatasız hale getirilmesi için harcanan süreler

Yöntem	Düzeltilme Süresi (dakika)
ANN-3	17 dakika
SOM – ANN-3	8 dakika

Çizelge 5. 2' deki değerler incelendiğinde, uygulanan yöntemlerin karşılaştırma sonuçlarıyla doğru orantılı olarak değiştiği gözlemlenmiştir.

Birinci, ikinci ve üçüncü aşamada kullanılan 6 numaralı görüntüden elle yapılan sayısallaştırma 37 dakika sürmüştür. Ayrıca her bir görüntünün DSAS analizi 1 saat sürmüştür.

Uygulama bölümünde anlatılan test aşamasında, SOM – ANN-3 yöntemine ait yapay sinir ağı, 6 farklı uydu görüntüsü kullanılarak test edilmiştir. Bu görüntülerden üç tanesi Terkos bölgesine ait farklı tarihli uydu görüntüleri, diğer üç tanesi ise İzmir, Antalya ve Van bölgelerine ait görüntülerdir. Bu görüntülerden elde edilen test sonuçları elle yapılan sayısallaştırmaları ile karşılaştırılarak Şekil 5. 5' deki sonuçlar elde edilmiştir.



Şekil 5. 5 Test aşamasına ait DSAS sonuçları

Terkos bölgesinin 2017, 2015 ve 2013 yıllarına ait test görüntülerinden elde edilen sonuçların ortalama hata değerleri sırasıyla 10,90 m (0,36 piksel), 10,80 m (0,36 piksel) ve 13,98 m (0,47 piksel) şeklindedir. İzmir ve Antalya bölgelerine ait test görüntülerinden elde edilen sonuçların ortalama hata değerleri sırasıyla 9,26 m (0,31 piksel) ve 39,31 m (1,31 piksel) şeklindedir. Erçek Gölü' ne ait test görüntüsünün sonucunun ortalama hata değeri ise 16,99 m (0,57 piksel) şeklindedir.

Bu sonuçlar incelendiğinde, SOM – ANN-3 yönteminin aynı bölgeye ait farklı zamanlı uydu görüntülerinden yüksek doğruluğa sahip ve yaklaşık olarak aynı sonuçlar ürettiği gözlemlenmiştir. Karadeniz Bölgesi' ne ait uydu görüntülerinden otomatik toplanan eğitim verileri ile eğitilen SOM – ANN-3 ağı, İzmir bölgesinden ve Van bölgesinden seçilen test alanlarında da yüksek doğruluğa sahip sonuçlar ortaya koymuştur. Fakat Antalya bölgesinden seçilen test alanında doğruluk yüksek olmamakla birlikte, yaklaşık 1,5 piksel olarak bulunmuştur.

Test aşamasındaki görüntülerin ağda işlenip kıyı çizgisi elde edilmesi 15 dakika sürmüştür. Ayrıca her bir görüntünün DSAS analizi yaklaşık 1 saat sürmüştür. Test görüntüleri üzerinden elle gerçekleştirilen sayısallaştırmalarının süreleri ise Çizelge 5. 3' de gösterilmiştir.

Çizelge 5. 3 Test görüntülerinden elle yapılan sayısallaştırmalar için harcanan zamanlar

Görüntü İsmi	Görüntü Numarası	Elle Sayısallaştırma Süresi (dakika)
Terkos 2017	7	35 dakika
Terkos 2015	8	36 dakika
Terkos 2013	9	34 dakika
İzmir 2017	10	52 dakika
Antalya 2017	11	47 dakika
Erçek Gölü 2017	12	59 dakika

Test sonucu elde edilen kıyı çizgilerinin kullanıcı tarafından düzeltilerek hatasız hale getirilmesi için harcanan süreler Çizelge 5. 4 'de gösterilmiştir.

Çizelge 5. 4 Test sonucu elde edilen kıyı çizgilerinin hatasız hale getirilmesi için harcanan süreler

Görüntü İsmi	Görüntü Numarası	Düzeltilme Süresi (dakika)
Terkos 2017	7	17 dakika
Terkos 2015	8	17 dakika
Terkos 2013	9	22 dakika
İzmir 2017	10	16 dakika
Antalya 2017	11	69 dakika
Erçek Gölü 2017	12	25 dakika

Çizelge 5. 4' deki değerler incelendiğinde, sürelerin karşılaştırma sonuçlarıyla doğru orantılı olarak değiştiği gözlemlenmiştir.

Sunulan tez kapsamında, uydu görüntülerinden yapay zekâ algoritmaları kullanılarak otomatik kıyı çizgisi çıkarımları gerçekleştirilmiştir. Otomatik olarak çıkarılan kıyı çizgileri, manuel olarak sayısallaştırılan kıyı çizgileri ile karşılaştırılarak tasarlanan yapay zekaların doğrulukları test edilmiştir. Ayrıca kıyı çizgilerinin elde edilme süreleri zamansal olarak da değerlendirilmiştir. Bu sayede yapay zekâ ile elde edilen kıyı çizgilerinin ne derece tutarlı ve doğru olduğu gözlemlenmiştir.

Bununla yanında bu çalışma ile yapay zekâ uygulamalarında kullanıcı tarafından toplanan eğitim verilerinin otomatik olarak toplanması sağlanarak, hem kullanıcı kaynaklı hatalar elimine edilmiş hem de zaman konusunda tasarruf sağlanmıştır.

## KAYNAKLAR

---

- [1] Awad, M., (2010). "An Unsupervised Artificial Neural Network Method for Satellite Image Segmentation", The International Arab Journal of Information Technology, 7(2): 199-205.
- [2] Yuan, H., Wiele, C. F. V. D. ve Khorram, S., (2009). "An Automated Artificial Neural Network System for Land Use/Land Cover Classification from Landsat TM Imagery", Remote Sensing Journal, 1: 243-265.
- [3] Marandi, R. J., Khanzadeh, M., Smith, B. K. ve Bian L., (2017). "Self-Organizing and Error Driven (SOED) artificial neural network for smarter classifications", Journal of Computational Design and Engineering, 4: 282-304.
- [4] Akar, Ö. ve Güngör, O., (2013). "Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması", Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi, 1(2):139-146.
- [5] Pal, M., (2013). "Random forest classifier for remote sensing classification", International Journal of Remote Sensing, 26(1): 217-222.
- [6] Bayram, B., Erdem, F., Akpınar, B., Ince, A., Bozkurt, S., Catal Reis, H. ve D. Z. Seker, (2017). "The Efficiency of Random Forest Method for Shoreline Extraction from Landsat-8 and Gokturk-2 Imageries", 4th International GeoAdvances Workshop, Karabuk, Turkey.
- [7] Bayram, B., Sen, A., Selbesoglu, M. O., Varna, I., Petersons, P., Aykut, N. O. ve Seker, D. Z, (2017). " Comparison of Two Simplification Methods for Shoreline Extraction from Digital Orthophoto Images", 4th International GeoAdvances Workshop, Karabuk, Turkey.
- [8] Çatal Reis H. ve Bayram, B., (2016). "Parçacık Sürü Optimizasyonu (Pso) Yöntemi ile Kıyı Çizgisi Çikartılması: Erçek Gölü Örneği", 6. Uzaktan Algılama-Cbs Sempozyumu (UZAL-CBS 2016), Adana, Türkiye.
- [9] Aydın, M. ve UANNI, M., (2013). " Kıyı Çizgisi Değişiminin Uydu Görüntüleri Yardımıyla İzlenmesi: Sakarya- Karasu", Harita Teknolojileri Elektronik Dergisi, 5(3): 24-32.
- [10] Kuleli, T., (2010). "Kızılırmak ve Yeşilirmak Deltalarındaki Kıyı Değişimlerinin Landsat TM ve DSAS ile Belirlenmesi", Türkiye' nin Kıyı ve Deniz Alanları VIII. Ulusal Kongresi, Trabzon, Türkiye.



- [11] Doğan, İ., (2008). Uzaktan Algılama Verileri İle Kıyı Çizgisi Değişiminin Zamansal Olarak Belirlenmesi: Alaçatı Örneği, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [12] Upadhyay, P. ve Kumar Chhabra, J., (2015). "Modified Self Organizing Feature Map Neural Network (MSOFM NN) Based Gray Image Segmentation", *Procedia Computer Science Journal*, 54: 671-675.
- [13] Pardo-Pacual, J. E., Almonacid-Caballer, J., A. Ruiz, L. ve Palomar-Vázquez, J., (2012). "Automatic extraction of shorelines from Landsat TM and ETM+ multi-temporal images with subpixel precision", *Remote Sensing of Environment Journal*, 123: 1-11.
- [14] Acar, U., Bayram, B., Balık Şanlı, F., Abdikan, S., Üstüner, M., Maktav, D. ve Sunar, F., (2012). "SAR GÖRÜNTÜLERİNDEN KIYI ŞERİDİ BELİRLEME ALGORİTMASI", IV. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu (UZAL-CBS 2012), Zonguldak, Türkiye.
- [15] Olgun, A., (2012). Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Yöntemiyle Göksu Deltası Kıyı Çizgisi Değişiminin İzlenmesi, Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [16] Kalkan, K., Mercan, O. Y. ve Maktav, D., (2010). "Kıyı Çizgisinin Otomatik Olarak Belirlenmesi ve Coğrafi Bilgi Sistemleri ile Entegrasyonu (Matlab Uygulaması)", III. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu, Kocaeli, Türkiye.
- [17] Sesli, F. A., (2006). "Sayısal Fotogrametri ile Kıyı Alanlarındaki Değişimin İzlenmesi", *Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi*, 95:11-17.
- [18] Ince, A., Bozkurt, S. ve Bayram, B., (2017). "Rasat Uydu Görüntüsünden Nesne Tabanlı Kıyı Çizgisi Çıkartma: Sapanca Gölü Örneği", 16. Türkiye Harita Bilimsel ve Teknik Kurultayı, Ankara, Türkiye.
- [19] Schowengerdt, R. A., (1997). *Remote Sensing: Models and Methods for Image Processing*, Second Edition, Academic Press, USA, ISBN: 0-12-628981-6.
- [20] Uzun, M., (2014). "Hersek Deltasında (Yalova) Kıyı Çizgisi-Kıyı Alanı Değişimleri ve Etkileri", *Doğu Coğrafya Dergisi*, 19(32): 27-32.
- [21] İnce, A., Bozkurt, S. ve Bayram, B., (2017). "RASAT Uydu Görüntüsünden Nesne Tabanlı Kıyı Çizgisi Çıkartma: Sapanca Gölü Örneği", 16. Türkiye Harita Bilimsel Teknik Kurultayı, 3-6 Mayıs 2017, Ankara.
- [22] Şen, A., (2013). Kartografik Genelleştirmede Seçme/Eleme İşlemi İçin Yapay Zekâ Yöntemlerinin Akarsu Ağlarına Uygulanabilirliği, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [23] Shyam M. Guthikonda, (2005), Kohonen Self-Organizing Maps, <http://www.shy.am/wp-content/uploads/2009/01/kohonen-self-organizing-maps-shyam-guthikonda.pdf>, 18 Ocak 2018.
- [24] Tom Germano, (1999), "Self Organizing Maps", <http://davis.wpi.edu/~matt/courses/soms/#Components>, 18 Ocak 2018.

- [25] Kohonen's Self Organizing Feature Maps, <http://www.ai-junkie.com/ann/som/som1.html>, 18 Ocak 2018.
- [26] Matsushita, H. ve Nishio, Y., (2010). "Batch-Learning Self-Organizing Map with Weighted Connections Avoiding False-Neighbor Effects", WCCI 2010 IEEE World Congress on Computational Intelligence, 18-23 July, Barcelona, Spain.
- [27] Nöcker, M., Mörchen, F. ve Ultsch, A., "An algorithm for fast and reliable ESOM learning", [https://www.academia.edu/32138210/An\\_algorithm\\_for\\_fast\\_and\\_reliable\\_ESOM\\_learning](https://www.academia.edu/32138210/An_algorithm_for_fast_and_reliable_ESOM_learning), 18 Ocak 2018.
- [28] Vesanto, J., (2002). Data Exploration Process Based on the Self-Organizing Map, Acta Polytechnica Scandinavica, Finnish Academies of Technology, Espoo, Finland.
- [29] Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V. ve Saarela, A., (2000). "Self Organization of a Massive Document Collection", IEEE Transactions On Neural Networks, 11(3): 574-585.
- [30] Rojas, R., (1996). Neural Networks – A System Introduction, Springer-Verlag, Berlin.
- [31] Cebeci, D., (2013). Kurumsal Kredi Değerlendirmede Bulanık Ahp-Yapay Sinir Ağları Temelli Bir Yaklaşım ve Bir Uygulama Çalışması, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [32] Öztemel, E., (2012). Yapay Sinir Ağları, 3. Baskı, Papatya Yayıncılık, İstanbul.
- [33] Sharma, B. ve Venugopalan, K., (2014). "Comparison of Neural Network Training Functions for Hematoma Classification in Brain CT Images", IOSR Journal of Computer Engineering, 16(1):31-35.
- [34] W.W. Hager ve H. Zhang, "A survey of nonlinear conjugate gradient methods", Pacific of Journal Optimization, 2(35):35–58.
- [35] Doan, C. D. ve Liong, S. Y., "Generalization For Multilayer Neural Network Bayesian Regularization Or Early Stopping", <http://masters.donntu.org/2012/etf/snegina/library/arl.pdf>, 18 Ocak 2018.
- [36] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning", Neural Networks, 6(4):525–533.
- [37] Elizabeth Conger, Line search methods, [https://optimization.mccormick.northwestern.edu/index.php/Line\\_search\\_methods](https://optimization.mccormick.northwestern.edu/index.php/Line_search_methods), 18 Ocak 2018.
- [38] Gill, E. P. ve Leonard, M. W., (2001). "Reduced-Hessian Quasi-Newton Methods for Unconstrained Optimization", SIAM Journal on Applied Mathematics, 12(1):209-237.
- [39] Pham, D. ve Sagirolu, S., (2001). "Training multilayered perceptrons for pattern recognition: a comparative study of four training algorithms", International Journal of Machine Tools and Manufacture, 41:419–430.

- [40] İbrahim Çayıroğlu, Ders Notları, Karabük Üniversitesi, <http://www.ibrahimcayiroglu.com/dokumanlar/ilerialgoritmaanalizi/ilerialgoritmaanalizi-5.hafta-yapaysiniraglari.pdf>, 18 Ocak 2018.
- [41] Bataineh, M. H., (2012). Artificial Neural Network for Studying Human Performance, Master of Science Degree Thesis, Iowa University, USA.
- [42] M., Forouzanfar, H. R., Dajani, V. Z., Groza, M., Bolic ve S., Rajan, (2010). "Comparison of Feed-Forward Neural Network Training Algorithms for Oscillometric Blood Pressure Estimation", 4th Int. Workshop Soft Computing Applications (SOFA 2010), July 2010, Arad, Romania ,119-23,
- [43] Chiang, Y. M., Chang, L. C. ve Chang, F. J., (2004). "Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling", Journal of Hydrology 290:297–311.
- [44] Rutka, G., (2006). "Neural Networks Models for Internet Traffic Prediction", Electronics and Electrical Engineering. - Kaunas: Technologija, 4(68):55–58.
- [45] Rahsepar, M. ve Mahmoodi, H., (2014). "Predicting Weekly Discharge Using Artificial Neural Network (ANN) Optimized By Artificial Bee Colony (ABC) Algorithm: A Case Study", Civil Engineering and Urban Planning: An International Journal (CiVEJ), 1(1):1-13.
- [46] Dorofki, M., Elshafie, A. H., Jaafar, O., Karim, O. A. ve Mastura, S., (2012). "Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data", 2012 International Conference on Environment, Energy and Biotechnology IPCBEE,33:39-44.
- [47] Ferhat Bağış, Yapay Sinir Ağlarına Giriş, <https://ferhatbagis.blogspot.com.tr/2017/05/yapay-sinir-aglarna-giris.html>, 18 Ocak 2018.
- [48] Bolat, S. ve Kalenderli, Ö., (2003). "Yapay Sinir Ağı ile İzolatör Konum Açısı Optimizasyonu", Elektrik-Elektronik-Bilgisayar Mühendisliği 10. Ulusal Kongresi, 189-192.
- [49] Yüksek, A. G., Bircan, H., Zontul, M. ve Kaynar, O., (2007). "Sivas İlinde Yapay Sinir Ağları ile Hava Kitlesi Modelinin Oluşturulması Üzerine Bir Uygulama", C.Ü. İktisadi ve İdari Bilimler Dergisi, 8(1):97-112.
- [50] Yazıcıoğlu, N., (2010). Yapay Zekâ ile Talep Tahmini, Yüksek Lisans Tezi, Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Bursa.
- [51] Civalek, Ö. ve Ülker, M., (2004). "Dikdörtgen Plakların Doğrusal Olmayan Analizinde Yapay Sinir Ağı Yaklaşımı", İMO Teknik Dergi, 3171-3190.
- [52] Umut Orhan, Ders Notları, Çukurova Üniversitesi, <http://ceng.cu.edu.tr/uorhan/DersNotu/Ders08.pdf>, 18 Ocak 2018.
- [53] Kavzoğlu, T. ve Çölkesen, İ., (2010). "Karar Ağaçları İle Uydu Görüntülerinin Sınıflandırılması: Kocaeli Örneği", Harita Teknolojileri Elektronik Dergisi, 2(1):36-45.

- [54] Akar, Ö., (2013). Rastgele Orman Sınıflandırıcısına Doku Özellikleri Entegre Edilerek Benzer Spektral Özellikteki Tarımsal Ürünlerin Sınıflandırılması, Doktora Tezi, KTÜ Fen Bilimleri Enstitüsü, Trabzon.
- [55] Sezer, E. A., Bozkır, A. S., Yağız, S. ve Gökçeoğlu C., (2010). “Karar Ağacı Derinliğinin CART Algoritmasında Kestirim Kapasitesine Etkisi: Bir Tünel Açma Makinesinin İlerleme Hızı Üzerinde Uygulama”, Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, 21-24 Haziran, İstanbul.
- [56] Tso, B. ve Mather, P. M., (2009). Classification Methods for Remotely Sensed Data, Second Edition, CRS Press, USA, ISBN 13: 978-1-4200-9072-7.
- [57] Breiman, L. ve Cutler, A., Random Forests, [https://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm), 18 Ocak 2018.
- [58] <https://woodshole.er.usgs.gov/project-pages/DSAS/>, 18 Ocak 2018.
- [59] Gencer, S., (2007). “Haritacılıkta Alan Hata Sınırı Formülleri ve İrdelenmesi”, Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi, 97:47-52.
- [60] [http://www.yapi.com.tr/haberler/karadeniz-az-daha-terkosu-yutacakti\\_28604.html](http://www.yapi.com.tr/haberler/karadeniz-az-daha-terkosu-yutacakti_28604.html), 18 Ocak 2018.
- [61] <https://landsat.gsfc.nasa.gov/landsat-8/landsat-8-bands/>, 18 Ocak 2018.
- [62] Hamarsheh, D., [http://www.philadelphia.edu.jo/academics/ghamarsheh/uploads/Lecture%2016\\_Self-organizing%20map%20using%20matlab.pdf](http://www.philadelphia.edu.jo/academics/ghamarsheh/uploads/Lecture%2016_Self-organizing%20map%20using%20matlab.pdf), 18 Ocak 2018.
- [63] <https://www.mathworks.com/help/nnet/ug/cluster-with-self-organizing-map-neural-network.html>, 18 Ocak 2018.
- [64] Kohonen, T., (2013). “Essentials of the self-organizing map”, Neural Networks, 37:52-65.
- [65] Yavuz, S. ve Devecei, M., (2013). “İstatistiksel Normalizasyon Tekniklerinin Yapay Sinir Ağın Performansına Etkisi”, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 40:167-187.
- [66] <https://www.mathworks.com/help/nnet/ref/dividetrain.html>, 18 Ocak 2018.
- [67] <https://www.mathworks.com/help/nnet/ref/dividerand.html>, 18 Ocak 2018.
- [68] Breiman L., (2002), Manual On Setting Up, Using, and Understanding Random Forests V3.1, [https://www.stat.berkeley.edu/~breiman/Using\\_random\\_forests\\_V3.1.pdf](https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf), 18 Ocak 218.

## ÖZGEÇMİŞ

---

### KİŞİSEL BİLGİLER

**Adı Soyadı** Abdulkadir İNCE  
**Doğum Tarihi ve Yeri** 8/3/1994 – İSTANBUL  
**Yabancı Dili** İNGİLİZCE, İSPANYOLCA  
**E-posta** [abdulkadir.ince34@gmail.com](mailto:abdulkadir.ince34@gmail.com)

### ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Harita Mühendisliği	Yıldız Teknik Üniversitesi	2016
Lise	Fen Bilimleri	İbn-i Sina Anadolu Lisesi	2012

### İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2015	İşlem CBS Mühendislik	Stajer Mühendis
2014	Dünya Harita	Stajer Mühendis

## YAYINLARI

### Makale

1. B. Bayram, F. Erdem, B. Akpınar, **A. K. Ince**, S. Bozkurt, H. Catal Reis, D. Z. Seker (2017), THE EFFICIENCY OF RANDOM FOREST METHOD FOR SHORELINE EXTRACTION FROM LANDSAT-8 AND GOKTURK-2 IMAGERIES, 4th International GeoAdvances Workshop, 14–15 October 2017, Safranbolu, Karabuk, Turkey.

### Bildiri

1. B. BAYRAM, N. DEMİR, D. Z. ŞEKER, M. OĞURLU, S. OY, S. BOZKURT, **A. İNCE** (2016), Mean-Shift Yöntemi İle Lidar – Intensity Verilerinden Kıyı Çizgisi Çıkartılması, Dünya CBS Günü, 24-25 Kasım 2016, İstanbul, Türkiye.
2. B. Bayram, N. Demir, B. Akpınar, **A. İnce**, S. Bozkurt, S. Oy, D. Z. Şeker, A. Güler, Y. Yüksel (2017), İha ve İha-Lidar Verilerinden Otomatik Üç Boyutlu Kıyı Çizgisi Çıkartılması Ve Analizi: Terkos Örneği, 16. Türkiye Harita Bilimsel ve Teknik Kurultayı, 3-6 Mayıs 2017, Ankara, Türkiye.
3. **A. İnce**, S. Bozkurt, B. Bayram (2017), Rasat Uydu Görüntüsünden Nesne Tabanlı Kıyı Çizgisi Çıkartma: Sapanca Gölü Örneği, 16. Türkiye Harita Bilimsel ve Teknik Kurultayı, 3-6 Mayıs 2017, Ankara, Türkiye
4. D. Z. Seker, N. Demir, S. Oy, **A. İnce**, M. Ogurlu, S. Bozkurt, B. Bayram (2017), An automated shoreline extraction approach using RASAT MS and SENTINEL 1A SAR images, 28. International Cartographic Conference, 2-7 July 2017, Washington D. C., USA.
5. N. Demir, B. Bayram, D. Z. Seker, S. Oy, S. Bozkurt, **A. İnce** (2017), Analyzing the Impact of an Earthquake on the Shoreline with use of SAR Images as Alternative Dataset, 19th International Symposium on Environmental Pollution and its Impact on Life in the Mediterranean Region, 4-6 October 2017, Rome, Italy.

### Proje

TÜBİTAK Projesi (115Y718)	Sürdürülebilir Kıyı Alanı İzleme Modeli İçin İha Ve İha-Lidar Verilerinden Otomatik Üç Boyutlu Kıyı Çizgisi Çıkartılması Ve Analizi: Terkos (İstanbul) Örneği
------------------------------	--