



**T.C.**  
**KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ELM ve DİĞER ALGORİTMALARLA**  
**3D SEGMENTASYON İŞLEMİ**

**ERTUĞRUL KAYA**

**YÜKSEK LİSANS TEZİ**  
**ENFORMATİK ANABİLİM DALI**

**KAHRAMANMARAŞ 2018**

**T.C.**  
**KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ELM VE DİĞER ALGORİTMALARLA**  
**3D SEGMENTASYON İŞLEMİ**

**ERTUĞRUL KAYA**

**Bu tez**  
**Enformatik Anabilim Dalında**  
**YÜKSEK LİSANS**  
**derecesi için hazırlanmıştır.**

**KAHRAMANMARAŞ 2018**

Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü öğrencisi Ertuğrul KAYA tarafından hazırlanan “ELM VE DİĞER ALGORİTMALARLA 3D SEGMENTASYON İŞLEMİ” adlı bu tez, jürimiz tarafından 19/10/2018 tarihinde oy birliği ile Enformatik Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. İbrahim Taner OKUMUŞ (ÜYE) .....  
Bilgisayar Müh.  
Kahramanmaraş Sütçü İmam Üniversitesi

Dr. Öğr. Üyesi Eser SERT (DANIŞMAN) .....  
Bilgisayar Müh.  
Kahramanmaraş Sütçü İmam Üniversitesi

Dr. Öğr. Üyesi Seda Aslan TUNCER(ÜYE) .....  
Yazılım Mühendisliği Bölümü  
Fırat Üniversitesi

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Doç. Dr. Mustafa ŞEKKELİ  
.....

Fen Bilimleri Enstitüsü Müdürü

## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada, alıntı yapılan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Ertuğrul KAYA

Bu çalışma Kahramanmaraş Sütçü İmam Üniversitesi, Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından desteklenmiştir.

Proje No: 2016/5-11 YLS

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

**ELM VE DİĞER ALGORİTMALARLA  
3D SEGMENTASYON İŞLEMİ  
(YÜKSEK LİSANS TEZİ)**

**Ertuğrul KAYA**

**ÖZET**

Günümüzde görüntü işlemeye tıp, askeri, eğitim, güvenlik gibi birçok alanda ihtiyaç duyulmaktadır. Segmentasyon işlemi de görüntü işlemenin en önemli adımlarından birini oluşturmaktadır. Literatür araştırmalarında, genellikle segmentasyon işleminin 2D olarak yapıldığı ve 3D segmentasyon çalışmalarının daha az olduğu görülmektedir. 3D segmentasyonda derinlik bilgisi de bulunduğundan, görüntünün daha doğru analiz edilmesi işlemine katkı sağlamaktadır. Bu durum 3D segmentasyon işleminin önemini daha da artırmaktadır.

Birçok alanda kullanılan ELM (Extreme Learning Machine – Aşırı Öğrenme Makinesi) tek gizli katmana sahip ileri beslemeli bir yapay sinir ağı modelidir. 3D segmentasyon işlemi için yapılan bu çalışmada ELM kullanılmıştır. Önerilen yöntemde ilk olarak ELM'nin eğitimi yapılmaktadır. Ardından test aşamasında, segmentasyonu istenen 3D görüntü kullanıcıya yardımcı olması için GUI (Graphical Unit Interface) kullanılarak seçtirilmektedir. Seçilen görüntüde daha önce eğitimi yapılan nesnelere var ise, tanınan nesnenin 3D olarak segmentasyonu gerçekleştirilmektedir. Buna ek olarak eğitim aşamasında nesnelere isimleri de öğretildiğinden, tanınan nesnelere yan taraflarına adları da yazdırılmaktadır. Segmentasyon için seçilen görüntüde eğitimi yapılmamış nesnelere var ise nesneyi ayrıştırarak yanına tanınmayan nesne ifadesini yazmaktadır. Önerilen yöntemle yapılan segmentasyon işleminin sonucu YSA ve FCM ile karşılaştırılmış olup elde edilen sonuçlar önerilen yöntemin başarısını açıkça ortaya koymuştur.

**Anahtar Kelimeler:** Segmentasyon, 3D Segmentasyon, ELM

Kahramanmaraş Sütçü İmam Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı, Ekim / 2018

Danışman: Dr. Öğr. Üyesi Eser SERT

Sayfa sayısı: 63

**3D SEGMENTATION WITH EXTREME LEARNING MACHINE AND  
OTHER ALGORITHMS  
(M.Sc. THESIS)**

**Ertuğrul KAYA**

**ABSTRACT**

Today, image processing is needed in many areas such as medicine, military, education and security. Segmentation is also one of the most important steps in image processing. In literature research, it is generally seen that the segmentation process is done in 2D and 3D segmentation studies are less. As 3D segmentation has depth information, it contributes to more accurate analysis of the image. This further increases the importance of 3D segmentation.

The ELM (Extreme Learning Machine), which is used in many areas, is a feed-forward neural network model with a single hidden layer. ELM is used in this study for 3D segmentation. In the proposed method, ELM is trained first. Then, during the testing phase, the segmented 3D image is selected using the GUI (Graphical Unit Interface) to assist the user. If there are objects previously trained in the selected image, 3D segmentation of the recognized object is performed. In addition, since the names of objects are also taught during the training phase, the names of the recognized objects are printed on their side. If there are objects that have not been trained in the image selected for segmentation, it parses the object and writes the unrecognized object expression next to it. The result of the segmentation process with the proposed method was compared with ANN and FCM, and the results clearly demonstrated the success of the proposed method.

**Key words:** Segmentation, 3D Segmentation, ELM

University of Kahramanmaraş Sütçü İmam  
Graduate School of Natural and Applied Sciences  
Department of Informatics, October / 2018

Supervisor: Dr. Öğr. Üyesi Eser SERT

Page Numbers:63

## TEŐEKKÜR

Bu tez alıőması sűresince engin bilgi ve tecrűbelerinden faydalandıđım ve alıőmamın her aőamasında sađladıđı bilimsel katkılardan dolayı Dr. Őđr. Őyesi Eser SERT'e teőekkűr ederim.

Bu tez alıőmamda manevi desteđini esirgemeyen eőim Ayőe Gűl'e, sabırlarından dolayı kızım Belinay'a ve ođlum Sayhan Eren'e sonsuz teőekkűrlerimi sunarım.

# İÇİNDEKİLER

## Sayfa No

ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ .....	vi
TABLolar DİZİNİ.....	viii
KISALTMALAR DİZİNİ .....	ix
1. GİRİŞ.....	1
1.1. Tezin Amacı .....	2
1.2. Literatür Araştırmaları .....	2
2. TEORİK ÇERÇEVE, MATERYAL VE YÖNTEM.....	3
2.1. Görüntü İşleme .....	3
2.2. Görüntü işleme ile ilgi temel kavramlar .....	3
2.2.1. Gri seviye resim.....	5
2.2.2. İkili(binary) resim.....	5
2.2.3. Renkli resim.....	5
2.2.4. MATLAB' de görüntü işleme amaçlı kullanılan önemli fonksiyonlar .....	6
2.2.5. PLY dosya formatı .....	7
2.2.6. Histogram işlemi .....	7
2.2.7. Histogram eşitleme .....	7
2.2.8. Görüntü eşikleme.....	8
2.2.9. Kenar belirleme (Edge detection).....	10
3. MAKİNE ÖĞRENMESİ YÖNTEMLERİ.....	12
3.1. Danışmanlı Öğrenme .....	12
3.2. Danışmansız Öğrenme.....	12
3.3. Destekli Öğrenme .....	12
4. SEGMENTASYON İŞLEMİ İÇİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ.....	13
4.1. Yapay Sinir Ağları.....	13
4.1.1. Biyolojik sinir hücresi .....	13
4.1.2. Yapay sinir hücresi .....	14
4.1.3. Girişler.....	14
4.1.4. Ağırlıklar .....	14
4.1.5. Toplama fonksiyonu .....	15
4.1.6. Aktivasyon fonksiyonu.....	16
4.1.6.1. Sigmoid fonksiyonu .....	17
4.1.6.2. Lineer (Doğrusal) fonksiyon .....	17
4.1.6.3. Hiperbolik tanjant fonksiyonu.....	18



4.1.6.4.	Step fonksiyonu .....	18
4.1.6.5.	Sinus fonksiyonu .....	19
4.1.6.6.	Eşik deęer fonksiyonu .....	19
4.1.7.	Hücrenin çıktısı .....	19
4.2.	İleri Beslemeli Yapay Sinir Aęı .....	20
4.3.	Geri Beslemeli Yapay Sinir Aęı .....	20
4.4.	K-Means .....	21
4.5.	FCM (Fuzzy c-means) .....	22
4.6.	ANFIS (Adaptive Neuro Fuzzy Inference System) .....	23
4.7.	Destek Vektör Makinesi (SVM - Suport Vector Machine) .....	25
4.7.1.	Doęrusal destek vektör makinesi .....	26
4.7.2.	Doęrusal olmayan destek vektör makinesi .....	26
4.8.	Aşırı Öğrenme Makinesi (ELM - Extreme Learning Machine ) .....	29
5.	3D SEGMENTASYON İŞLEMİ İÇİN ÖNERİLEN YÖNTEM .....	32
5.1.	ELM'nin Eğitimi .....	34
5.2.	ELM'nin Test Bölümü .....	38
5.3.	ELM ile Segmentasyon İçin Hazırlanan Yazılım .....	45
5.4.	Kullanılan Donanımlar .....	48
5.5.	Deneysel Çalışmalar .....	49
6.	SONUÇ .....	60
	KAYNAKLAR .....	61
	ÖZGEÇMİŞ .....	63

## ŞEKİLLER DİZİNİ

	<b><u>Sayfa No</u></b>
Şekil 2.1 Görüntünün sayısallaştırılması prensibi .....	4
Şekil 2.2 sayısal görüntü oluşması .....	5
Şekil 2.3 Örnek bir resim ve histogramı .....	8
Şekil 2.4 Histogramı eşitlenmiş bir resim ve histogramı .....	8
Şekil 2.5 Orijinal görüntü .....	9
Şekil 2.6 Eşiklenmiş görüntü .....	9
Şekil 2.7 Sobel hazırlamak için maskeler, (a) gradient $G_x$ ve (b) gradient $G_y$ .....	10
Şekil 3.1 Danışmanlı öğrenme mimarisi .....	12
Şekil 4.1 Biyolojik sinir hücresinin yapısı .....	14
Şekil 4.2 Yapay sinir hücresinin yapısı .....	15
Şekil 4.3 Sigmoid fonksiyonu .....	17
Şekil 4.4 Lineer fonksiyonu .....	18
Şekil 4.5 Hiperbolik tanjant fonksiyonu .....	18
Şekil 4.6 İleri beslemeli yapay sinir ağı modeli .....	20
Şekil 4.7 Geri beslemeli yapay sinir ağı modeli .....	21
Şekil 4.8 İki girişli anfis mimarisi .....	23
Şekil 4.9 Destek vektör makinesi .....	25
Şekil 4.10 Doğrusal ayırlamama durumu .....	25
Şekil 4.11 a) Doğrusal olmayan veri setinin 3 boyutlu uzaya taşınarak sınıflandırılması ..	28
Şekil 4.11 b) Doğrusal olmayan veri setinin 3 boyutlu uzaya taşınarak sınıflandırılması ..	28
Şekil 4.12 Tek gizli katmana sahip ileri beslemeli bir YSA .....	289
Şekil 4.13 Tek gizli katmana sahip ELM modeli .....	30
Şekil 5.1 Önerilen yöntemle eğitim sürecine ait akış diyagram .....	33
Şekil 5.2 Önerilen yöntemle test sürecine ait akış diyagram .....	34

**Sayfa No**

Şekil 5.3 Orijinal görüntü .....	36
Şekil 5.4 Ön işlemden sonra elde edilen görüntü .....	36
Şekil 5.5 Görüntünün orijinali .....	37
Şekil 5.6 Ön işlem sonucu elde edilen görüntü .....	37
Şekil 5.7 Orijinal üç boyutlu görüntünün iki boyuta dönüştürülmüş şekli.....	41
Şekil 5.8 Eğitimi yapılmış nesnelerin segmentasyonu ve isimlerinin yazılması.....	424
Şekil 5.9 Önerilen yöntemle segmentasyon için hazırlanan gui.....	426
Şekil 5.10 3D görüntü almak için kullanılan cihaz .....	428
Şekil 5.11 Deneysel çalışmalardan görüntü .....	449
Şekil 5.12 Eğitim yapılan iki objenin segmentasyonu .....	50
Şekil 5.13 Eğitim yapılan iki objenin segmentasyonu .....	51
Şekil 5.14 Eğitimi yapılan obje ile eğitimi yapılmayan obje segmentasyon sonucu .....	51
Şekil 5.15 Karşılaştırmalı JI Grafiği.....	56
Şekil 5.16 3D model üzerinde ölçülendirmeler .....	56
Şekil 5.17 Yöntemlere göre karşılaştırmalı $\Delta X$ grafiği .....	58
Şekil 5.18 Yöntemlere göre karşılaştırmalı $\Delta Y$ grafiği .....	58
Şekil 5.19 Yöntemlere göre karşılaştırmalı $\Delta Z$ grafiği .....	59

## TABLULAR DİZİNİ

	<b><u>Sayfa No</u></b>
Tablo 4.1 Toplama fonksiyonları ve açıklamaları.....	156
Tablo 5.1 a) ELM, YSA, FCM segmentasyon sonuçları .....	522
Tablo 5.1 b) ELM, YSA, FCM segmentasyon sonuçları .....	523
Tablo 5.2 Üç yaklaşıma ait Jaccard Index sonuçları .....	55
Tablo 5.3 Jaccard Index sonuçlarına ait istatistiksel sonuçlar.....	55
Tablo 5.4 Hata farkları.....	57
Tablo 5.5 Hata farklarına ait istatistiksel sonuçlar .....	57

## KISALTMALAR DİZİNİ

- ELM** :Extreme Learning Machine (Aşırı Öğrenme Makinesi)  
**YSA** :Yapay Sinir Ağları  
**FCM** :Fuzzy C-Means  
**PLY** :Polygon File Format Stanford Üçgeni

## 1. GİRİŞ

Herhangi bir objeyi algılamak için bakıldığında görüntü gözün iç tarafındaki retinaya aktarılmaktadır. Retina milyonlarca görme hücresinden meydana gelir. Retinadan sonra görüntüler elektrik sinyallerine dönüştürülerek görme sinirleri tarafından beyne iletilir. Bu sinyaller beynin farklı bölgelerince işlenerek anlamlı hale getirilir. Böylece çevremizi görsel olarak algılama, çıkarım yapma ve iletişime geçme süreçleri gerçekleştirilmektedir. Benzer şekilde, bilgisayarla gerçekleştirilen görüntü işleme uygulamalarında da kameralar aracılığıyla alınan görüntüler belirli algoritmik süreçlerden geçirildikten sonra anlamlandırılmakta, çıkarımlar yapılmakta ve görsel modeller çıkarılmaktadır

Yapay sinir ağları (YSA) kendilerine verilen verileri işleyerek bir veri çıkışı üretirken, giriş örnekleri ile olaylar arasındaki ilişkileri öğrenen sistemlerdir. YSA' ların en önemli özelliklerinden birisi de, test aşamasında hiç görmediği veri girişi hakkında öğrendikleri bilgileri kullanarak karar verirken gerçek hayattaki gibi doğrusal olmayan yapıları da dikkate alıyor olabilmesidir. YSA' ların temel işlem elemanlarından olan hücre doğrusal değildir. Dolayısıyla bu hücrelerin birleşiminden meydana gelen YSA' larda doğrusal değildir[1]. Son zamanlarda popüler hale gelmiş olan YSA modellerinden ELM (Extreme Learning Machine) tek gizli katmana sahip ileri beslemeli bir yapay sinir ağı modelidir. Söz konusu modelde giriş ağırlıkları ve gizli katman ağırlıkları rastgele, çıkış ağırlıkları ise analitik olarak hesaplanmaktadır[2].

Günümüzde farklı kamera türlerinden görüntü elde edilebilmektedir. Bu görüntülerin bilgiye dönüştürülebilmesi için birçok işlemden geçirilmesi gerekmektedir. Bu işlemlerin ilk basamağını da genellikle segmentasyon işlemi oluşturmaktadır. Segmentasyon, bir görüntüyü her biri içerisinde farklı özelliklerin tutulduğu bölgelere ayırma olarak ifade edilebilir. 3D teknolojisinin 2D teknolojisinden farklı olarak derinlik bilgisi içerdiğinden, bu teknoloji nesnelerin sayısal sistemler tarafından daha doğru algılanmasını sağlamaktadır. Bu durum 3D segmentasyon işleminin önemini artırmaktadır. Bundan dolayı bu çalışmada ELM ile 3D modelleme işlemi gerçekleştirilmektedir.

## **1.1. Tezin Amacı**

Bu çalışmada ELM algoritması kullanılarak 3D görüntülerin segmentasyonu gerçekleştirilmektedir. ELM tek gizli katmana sahip ileri beslemeli bir yapay sinir ağı modelidir. Söz konusu modelde giriş ağırlıkları ve gizli katman ağırlıkları rastgele, çıkış ağırlıkları ise analitik olarak hesaplanmaktadır[2].

Segmentasyonda temel amaç istenen 3D tarayıcı cihazdan elde edilen 3D modelden yalnızca eğitimi yapılan nesne/nesnelerin 3D modelini çıkarmaktır. ELM eğitildikten sonra objeleri tanıyarak bir çıkarım yapabilmesi özellikle güvenlik ve askeri alanda etkin bir şekilde kullanımını ön plana çıkarmaktadır. Bu çalışmanın temel amacı tıp, askeri, güvenlik, eğitim gibi alanlarda ihtiyaç duyulan 3D segmentasyon ihtiyacının giderilmesine yardımcı olmaktır.

## **1.2. Literatür Araştırmaları**

Yapılan literatür araştırmalarında ELM algoritması ile gerçekleştirilen 3D segmentasyon çalışmalarının çok sınırlı olduğu görülmüştür. Sağlık alanında yapılan çalışmalarda da ELM algoritması sıklıkla kullanılmaktadır. Bu çalışmalara paralel olarak, Kaya ve Tekin tarafından ELM kullanılarak medikal tanı sistemlerinde kullanılabilecek bir uzman sistem tasarlanmıştır [3]. Ertuğrul tarafından Aşırı Öğrenme Makineleri ile sağlık alanında kullanılabilecek bir sistem tasarlanmıştır[4].

Zhang, Lu ve Marziliano tarafından yapılan çalışmada diz eklemlerinin manyetik rezonans (MR) görüntülerindeki menisküs tespiti için ELM algoritması ile segmentasyon işlemi yapılmıştır [5]. An ve Bhanu tarafından yapılan çalışmada ise fotoğraflar için en iyi çözünürlüğün tespiti için ELM algoritması kullanılmıştır [6]. Lu ve diğerleri tarafından ELM kullanılarak videolarda semantik kavram algılama çalışması yapılmıştır [7].

## **2. TEORİK ÇERÇEVE, MATERYAL VE YÖNTEM**

### **2.1. Görüntü İşleme**

Etrafımıza baktığımızda sürekli gördüğümüz bize sıradan gibi görünen nesnelere ve bu nesnelere şekilleri, renkleri, beynimizin algılaması için gözden içeri giren ışıkla birlikte çok hızlı olarak ve birçok işlemden geçmesi gerekmektedir. Retinadan beyne sinyal yaklaşık bir buçuk milyon nöron tarafından taşınır. Retina altıgen dizi şeklinde çok fazla foto reseptör içerir. Nesnelere beynimiz tarafından nesnelere kenarlarından tanınır. Nesnelere veya sahnenin içindeki bölgelere kaplayan sınırlar, nesne hakkında maksimum bilgi bu kenarlar vasıtasıyla elde edilir. Nesnelere tanıma süreci nöral organizasyonumuzda yer alan öğrenmenin sonucudur[8].

Görüntü işleminin temel amaçları, bir görüntünün dijital forma dönüştürülerek bu görüntüyü daha net hale getirmek, görüntü üzerinde istenilen değişikliğin yapılması veya görüntüden bilgiler elde ederek bu bilgilerin kullanılması için yapılması gereken işlemler olarak sıralanabilmektedir. Görüntü analiz sistemlerinin tasarlanması işleminde ilk adım dijital görüntünün elde edilmesidir. Günümüzde farklı birçok ekipmanlarla dijital görüntüler elde edilmektedir. Optik veya termal dalga boylarında sensörlere sahip cihazlar kullanılarak görüntüler elde edilmektedir. Elde edilen bu görüntülerin bilgisayar tarafından algılanıp değişik işlemlerden sonra anlamlı sonuçlar çıkarabilmesi için yapılacak ilk işlemlerden biri segmentasyon işlemi diyebiliriz.

Segmentasyon işlemi, bir görüntüyü birden fazla kendi içerisinde homojen bölgeye ayırma olarak tanımlanabilmektedir. Bölgedeki her bir piksel benzersiz bir bölge etiketi edinir. Otomatik görüntü işlemede, ilk olarak nesnelere bir sonraki işlem için görüntüden çıkarılır. Örneğin bir kuş ve kutu bulunan bir imgeyi işlemek için öncelikle imge bölümlere ayrılmalıdır. Sonra imgede bulunan nesnelere sınıflandırılmalıdır. Bölümler oluştuktan sonra doku, renk, şekil gibi özellikler kullanılarak bir dizi oluşturulur ve nesnelere imgeden çıkartılarak yapılmak istenen işleme tabi tutulur[8].

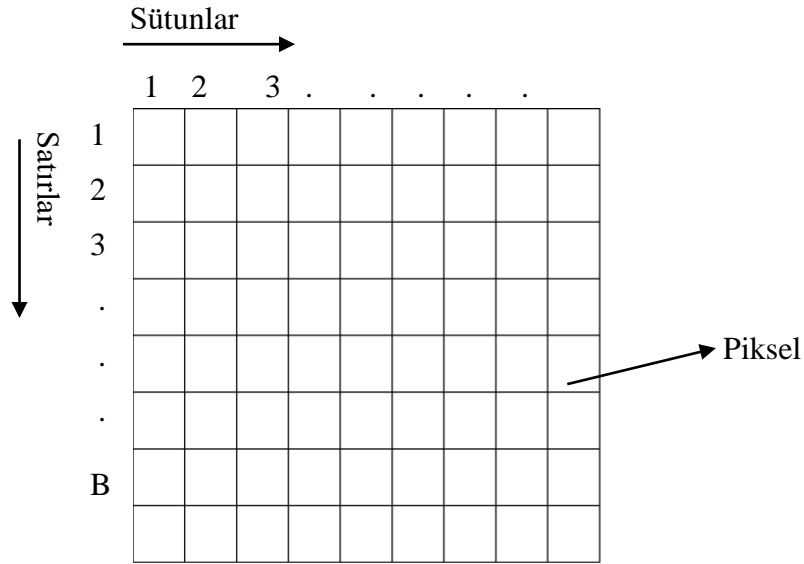
### **2.2. Görüntü işleme ile ilgili temel kavramlar**

Piksel, görüntünün oluşması için gerekli olan en küçük birime denir. Görüntüyü dijital ortamda işlemek, kontrol etmek içinde en küçük birim olan piksel kullanılır. Bilindiği üzere görüntüler piksellerden oluşur, çözünürlük görüntünün kaç pikselden oluştuğunu gösterir. Bir görüntüde çözünürlük ne kadar fazla ise görüntüdeki ayrıntılar o kadar belirginleşir.



Görüntü,  $k(A, B)$  gibi bir fonksiyonla ifade edilen bir imgede,  $k$  herhangi bir şiddet birimini mesela parlaklığı ifade ederken,  $A$  ve  $B$  ise görüntünün yatay dikey koordinatlarına karşılık gelen değişkenlerdir.  $k$ ,  $A$  ve  $B$  ayrık ve sonlu ise bu görüntü sayısal görüntü olarak ifade edilmektedir. Söz konusu imgenin  $A$  sütun ve  $B$  satırdan oluşacak şekilde örneklenmesi sonucu elde edilen görüntüye sayısal görüntü, satır ve sütunun kesiştiği alana piksel denir.

Elde edilen sayısal görüntüde  $A \times B$  adet piksel bulunur. Bu durum şekil'2.1 de gösterilmiştir. Görüntü sayısal görüntüye dönüştürüldükten sonra 1 değeri aydınlık yani beyaz bölgeleri, 0 değeri karanlık yani siyah bölgeleri temsil eder. Sayısal görüntüde renkli olan dosyalar genellikle 24 ya da 8 bit olurken, gri seviye görüntüler ise 1,2,4,6 veya 8 bit olabilirler.



Şekil 2.1 Görüntünün sayısallaştırılması prensibi

Sayısal bir görüntüye değer atanması ve saklanması örneği şekil 2.2'de gösterilmiştir. Örnekte T harfinin işlemi gösterilmiştir.

1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1

Şekil 2.2 sayısal görüntü oluşması

### 2.2.1. Gri seviye resim

Görüntü işlemede çoğu zaman işlem yapılacak renkli görüntü öncelikle gri formata dönüştürülür. MATLAB’ de bu dönüştürme işlemi için `rgb2gray` fonksiyonu kullanılmaktadır. Bu fonksiyon RGB (Red Green Blue-Kırmızı Yeşil Mavi) formatındaki renkli görüntüyü gri tonlamalı görüntüye dönüştürür. Bu dönüştürme işlemi R, G ve B bileşenlerinin ağırlıklı bir toplamını oluşturarak elde eder. Gri tonlama değerlerini hesaplamak için kullanılan katsayılar  $0.2989 * R + 0.5870 * G + 0.1140 * B$  şeklindedir [9].

### 2.2.2. İkili(binary) resim

İkili görüntü siyah ve beyaz dan oluşur. Gri tonlamalı görüntülerde piksel değerleri 0-255 aralığında bir değer olarak gri tonları oluşturur. İkili görüntülerde ise piksel değerleri sadece 0 ve 1’den oluşan değerler alırlar. Burada 0 siyah bölgeyi 1 ise beyaz bölgeyi temsil eder. Şekil 2.2’de ikili bir görüntü görülmektedir.

### 2.2.3. Renkli resim

Görünür renkler elektromanyetik spektrumda yaklaşık 400 nm(Mor) ve 700 nm(Kırmızı) arasında görünür. Renkli görüntülerde her pikselin rengi kırmızı, yeşil ve mavi yoğunlukların kombinasyonu ile belirlenir.[9] Görüntülerde kullanılan 8 bit formatı içinde 256 renk tonu vardır. Genellikle gri tonlamalı görüntülerde kullanılır. 0-256 arasında bir değer alır. Piksel 0-256 değer aralığında siyahtan başlayıp (0) beyaza doğru (256) bir değer alır. 16 bitlik renk formatı, içinde  $2^{16}$  yani 65.536 farklı rengi barındırır. 16 bit formatı, kırmızı, yeşil ve mavi kanallarında kırmızı için 5 bit, yeşil için 5 bit ve mavi için 5 bit olarak dağıtılır geriye kalan bir bit ise alfa kanalına ayrılır[10].

24 bitlik renk formatında, $2^{24}$  yani 16.777.216 farklı renk tonunu destekler bu da gerçek yaşamdaki bir görüntüyü oluşturabilir. Kırmızı, yeşil, mavi renk kanallarına eşit olarak 8 bit dağıtılmıştır. 32 bitlik renk formatında renk bilgilerini saklamak için 24 bit kullanır. Kalan 8 biti ise saydam verileri (alfa kanalı) saklamak için kullanır[10].

Renk modelleri, belirli bir rengi belirtmek için üç boyutlu bir koordinat sistemi tanımlayarak standart bir yol ve belirli bir modeldeki tüm yapılandırabilir renkleri içeren bir alt uzay sağlar. Bir model kullanılarak belirtilebilecek herhangi bir renk, tanımlandığı alt uzaydaki tek bir noktaya karşılık gelecektir. Renk modelleri genel olarak RGB, CMY olarak belirtilir[9].

RGB(Red Green Blue - Kırmızı, Yeşil, Mavi) modeli, bu modelde bir görüntü ana renklerin her birinde bir tane olmak üzere üç bağımsız görüntü düzleminden kırmızı, yeşil ve mavi den oluşur. Belirli bir rengin belirlenmesi mevcut ana bileşenlerin her birinin miktarını belirtmektedir[9].

CMY(Cyan Magenta Yellow – Cam göbeği, Eflatun, Sarı) modeli, RGB modeli ile belirli bir rengi elde etmek için siyahın neye eklendiği sorulurken, CMY modeli ile, beyazdan neyin çıkarıldığı sorulur. Bu durumda ilk renkler cam göbeği, eflatun ve sarı olurken ikincil renkler kırmızı, yeşil ve mavi renklerdir[9].

#### **2.2.4. MATLAB’ de görüntü işleme amaçlı kullanılan önemli fonksiyonlar**

MATLAB’ de kullanılan önemli bir fonksiyon olan graythresh, bir görüntüyü ikili bir resme dönüştürmede kullanılan fonksiyonlar için kullanılabilir global bir eşik değeri hesaplar. Elde edilen bu değer görüntüleri ikili formata, yani siyah beyaz görüntülere dönüştürmede kullanılan imbinarize fonksiyonuyla kullanılır. Graythresh fonksiyonu siyah ve beyaz piksel içi varyans değerini küçültmek için eşik değerini hesaplamada Otsu yöntemini kullanır[11]. Varyans, bilindiği üzere verilerin dağılımına ait her bir verinin değerinin tüm verilerin ortalamasından ne kadar uzakta olduğunu ifade eder. Otsu yöntemi, bir görüntüyü ikili bir görüntüye dönüştürme işleminde kullanılmak üzere 0 ile 1 aralığında bir değer üretir. Bu değeri siyah ve beyaz piksellerin varyansını en aza indirmek için bir eşik değeri olarak sunar[12].

*Otsu*, bir görüntüyü ikili bir görüntüye dönüştürme işleminde kullanılmak üzere 0 ile 1 aralığında bir değer üretir. Bu değeri siyah ve beyaz piksellerin varyansını en aza indirmek için bir eşik değeri olarak sunar[12].

Görüntü üzerinde istenen işlemler yapıldıktan sonra bazı durumlarda elde edilen görüntü dosyasının kaydedilmesi gerekebilir. Görüntü kaydetmek için kullanılan fonksiyon *imwrite* fonksiyonudur. Bu fonksiyonla görüntüler istenilen resim türlerinde(jpeg,png,gif) kaydedilebilir[13].

### **2.2.5. PLY dosya formatı**

PLY dosya formatı, 3D verilerinin saklanması için esnek ve sistematik bir program tanımlar. Stanford üçgen formatı olarak da bilinen PLY formatı ASCII üstbilgisi, her bir özellikler dizisiyle nesnelere tanımlayarak dosyada hangi verilerin olduğunu belirtir[14]. PLY formatı, bir başlıktan sonra bir köşe listesi ve daha sonra bir poligon listesi içerir.

Üstbilgi dosyada kaç tane köşe ve çokgen bulunduğunu belirtir ve ayrıca, her bir köşe (x,y,z) koordinatlar, normaller ve renk gibi hangi özelliklerin ilişkilendirildiğini belirtir. Çokgen yüzler, köşe listesindeki basitçe indis listesidir. Her bir yüz, her listedeki öğe sayısının bir sayısı ile başlar[15].

### **2.2.6. Histogram işlemi**

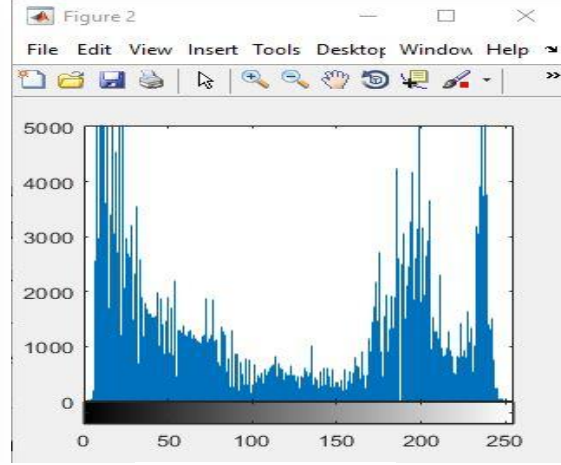
Görüntü işlemede yaygın olarak kullanılan histogram, bir görüntünün basit ama çok önemli istatistiksel bir özelliğidir. Yoğunluk histogramı görüntü içindeki tüm piksellerin gri seviye değerlerinin bir dağılımıdır. Histogramdaki her bölme yoğunluk değerleri söz konusu kutuda bulunan piksel sayısını temsil eder. Her gri seviyenin bir bölme karşılık geldiği 255 gri tonlu bir histogram kullanılır. Örnek bir manzara fotoğrafı şekil 2.3 a)'da ve fotoğrafın histogram grafiği şekil 2.3 b)'de gösterilmiştir[16].

### **2.2.7. Histogram eşitleme**

Histogram eşitleme, yoğunluk değerlerinin dönüştürülmesini içerir, böylece çıktı görüntüsünün histogramı, belirtilen bir histogram ile eşleşir. Histogram eşitlemenin temel fikri, dönüştürülmüş görüntünün histogramının tekdüze olacağı şekilde yoğunluk dönüşümünü bulmaktır. Yoğunluk değerlerini ayarlama işlemi, histogram dengelemesi kullanılarak otomatik olarak yapılabilir. Şekil 2.4 a)' da histogram eşitlemesi uygulanmış fotoğrafı, Şekil 2.4 b)' de histogram grafiği gösterilmiştir. İşlem sonucunda fotoğraftaki ağaçların daha net görüldüğü görülmektedir[16].



a) Manzara fotoğrafı

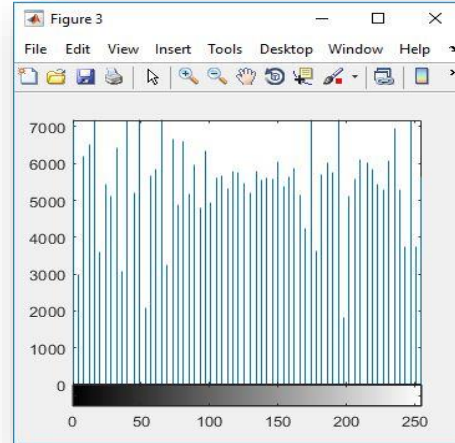


b) Histogram

Şekil 2.3 Örnek bir resim ve histogramı



a) Histogram eşitleme uygulanmış fotoğraf



b) Histogram eşitleme uygulanmış fotoğraf grafiği

Şekil 2.4 Histogramı eşitlenmiş bir resim ve histogramı

### 2.2.8. Görüntü eşikleme

Elde edilen bir görüntünün parlaklığını ve kontrastını değiştirmeye bağlı olarak objeleri arka planından ayırmak için kullanılan tekniklerden biridir. Objeleri arka plandan ayırmak için, histogramdan göreceli olarak belirlenen bir  $T$  eşik değeri ile elde edilen görüntüdeki piksel değerlerini karşılaştırarak, herhangi bir  $(i,k)$  pikseli için;  $f(i,k) > T$  ise  $(i,k)$  pikseli nesneye ait bir nokta,  $f(i,k) \leq T$  ise  $(i,k)$  pikseli arka plana ait bir nokta olacaktır. Şekil 2.5'te orijinal görüntü, şekil 2.6'da ise orijinal görüntünün eşikleme yapılmış hali görülmektedir[17].

Genel bir eşikleme fonksiyonu ve eşiklenmiş bir  $f(x,y)$  görüntüsünün denklemi denklem 2.1 de gösterilmiştir. Belirlenen piksellerden 1 nesneye, 0 ise arka plana denk gelir. T eşikleme fonksiyonunda;  $g(x,y)$ ,  $(x,y)$  noktasının gri seviye değerini,  $k(x,y)$  noktası ise bu noktanın bazı bölgesel özelliklerin değerini belirtir[13]. Örneğin bu özellik  $(x,y)$  noktasının komşuluğundaki piksel gri seviyelerin ortalaması olabilir.



Şekil 2.5 Orijinal görüntü



Şekil 2.6 Eşiklenmiş görüntü

T sadece  $g(x,y)$ ' ye bağlı ise, denklem 2.1 deki T eşitliği ile belirlenen eşik değeri bütünsel (Global) eşik olarak isimlendirilir. T,  $g(x,y)$  ve aynı zamanda  $k(x,y)$ 'ye bağlı ise bu durumda denklem 2.1 deki T eşik değeri bölgesel(local) eşik olarak isimlendirilir.

$$T = T[x, y, g(x, y), k(x, y)]$$
$$f(x, y) = \begin{cases} 1 & \text{eger } g(x,y) > T \\ 0 & \text{eger } g(x,y) \leq T \end{cases} \quad (2.1)$$

### 2.2.9. Kenar belirleme (Edge detection)

Görüntüdeki kenardan maksat görüntünün gri seviyelerindeki ani değişikliklerin olduğu bölgelerdir. Elde edilen görüntünün kenarlarının belirlenmesi obje tanıma, tanınan objeyi arka plandan ayırma, görüntüdeki objelerin boyutunun ölçülmesi gibi birçok görüntü işleme işlemlerinde önemli rol oynamaktadır. Kenar belirleme yöntemlerinin amaçlarından birisi de ani gri seviye farklılıklarını tespit etmektir. Bu doğrultudaki kenar belirleme yöntemlerindeki yaklaşım görüntünün bölgesel türev hesabıdır[16].

Türev kullanılarak kenar bulma algoritmalarından sobel, 3 X 3 komşu tabanlı bir gradyan operatörüdür. Sobel operatörünün konvülasyon maskeleri şekil 2.7’de gösterilen iki çekirdek tarafından tanımlanmıştır. İki maske, sırasıyla yatay ve dikey yönlerde iki gradyan bileşeni denklem 2.2’de gösterilen  $G_x$  ve  $G_y$  üretmek için giriş görüntüsüne ayrı ayrı uygulanır[8].

1	2	1
0	0	0
-1	-2	-1

(a)

1	0	-1
2	0	-2
1	0	-1

(b)

Şekil 2.7 Sobel hazırlamak için maskeler, (a) gradient  $G_x$  ve (b) gradient  $G_y$

$$G_x = [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] - [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)] \quad (2.2)$$

$$G_y = [f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)] - [f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1)]$$

Kenar algılama işlemi, bir görüntüdeki yoğunluk seviyesindeki önemli yerel değişiklikleri saptamak için bir işlemdir. Yoğunluk seviyesindeki değişim, görüntünün gradyanı ile ölçülür. Bir resim denklem 2.3’teki gibi,  $f(x, y)$  iki boyutlu bir işlev olduğundan, eğimi bir vektördür.

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad (2.3)$$

Degradenin büyüklüğü denklem 2.4’de gösterildiği gibi birkaç değişik şekilde hesaplanabilir[8].

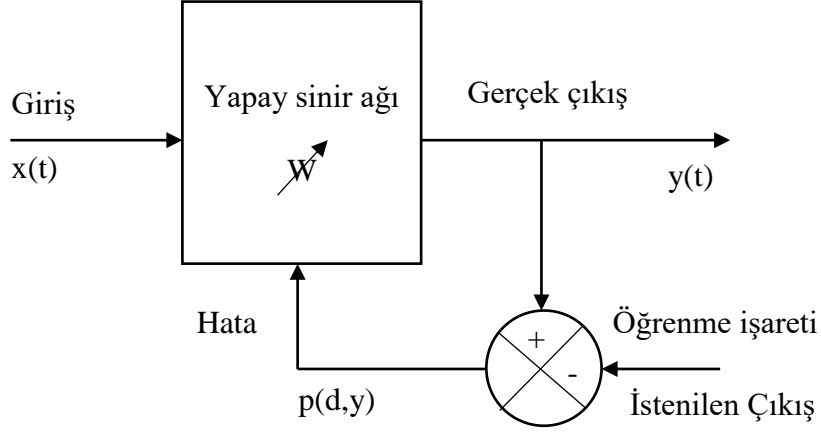
$$\begin{aligned} G[f(x, y)] &= \sqrt{G_x^2 + G_y^2} \\ G[f(x, y)] &= |G_x| + |G_y| \\ G[f(x, y)] &= \max \{ |G_x|, |G_y| \} \end{aligned} \quad (2.4)$$



### 3. MAKİNE ÖĞRENMESİ YÖNTEMLERİ

#### 3.1. Danışmanlı Öğrenme

Bu eğitim stratejisinde eğitim esnasında hem giriş verileri hem de bu girişlere ait üretilmesi gereken çıktılar verilir. Ağdan beklenen her giriş için o girişe karşılık gelen çıktıyı elde etmesidir. Ağ ürettiği çıkış değerini istenilen çıkış değeriyle karşılaştırarak ağırlıkları buna göre değiştirir. Girilen değerle istenilen değer arasındaki fark hatadır. Hata değeri önceden belirlenen değerden küçük oluncaya kadar eğitime devam edilir. Hata değeri istenen değer altına düştüğünde ağırlıkların tamamı sabitlenerek eğitim işlemi sonlandırılır. Eğer ağa gösterilen örneklere karşılık beklenen çıktı üretilmiyorsa ağın çıkış değerindeki hatayı en küçük değere çekecek şekilde bağlantı ağırlıkları değiştirilir [18]. Şekil 3.1’de danışmanlı öğrenme mimarisi görülmektedir.



Şekil 3.1 Danışmanlı öğrenme mimarisi

#### 3.2. Danışmansız Öğrenme

Bu öğrenme şeklinde sisteme, giriş verilerine karşılık gelecek bir çıkış verisi verilmemektedir. Bu teknikte girişine uygulanan bilgilere bağımlı olarak yöntem her bir örneği sınıflandıracak şekilde kural sistematığı oluşturmaktadır.

#### 3.3. Destekli Öğrenme

Bu tür öğrenme şekli danışmanlı öğrenmeye yakın bir öğrenme şeklidir. Yapay sinir ağına öğrenilmesi istenen veriler verilmekte çıktıyı ağın kendisinin üretmesi istenmektedir. Çıkış verileri alındıktan sonra, ağa verilen verilere karşılık elde edilen çıkışların doğru olup olmadığı söylenmektedir [18]

## 4. SEGMENTASYON İŞLEMİ İÇİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ

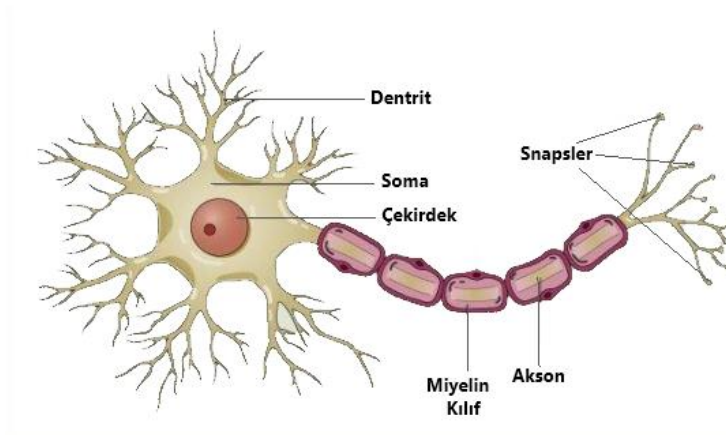
### 4.1. Yapay Sinir Ağları

Yapay sinir ağları, insan beyninin öğrenme ilkesini temel alarak, bilgisayarlar üzerinde taklit edilmesi fikri ile ortaya çıktığından öncelikle biyolojik sinir hücresini tanımak gerekmektedir. Bundan dolayı ilk aşamada biyolojik sinir hücresi hakkında bilgi verilmiştir.

#### 4.1.1. Biyolojik sinir hücresi

İnsan beyninin değişik bölgeleri değişik fonksiyonları yerine getirmektedir. İnsan vücudunun organlarından gelen sinyaller (bilgiler), sinir sistemi ile beyine ulaştırılır beyin verdiği kararlarda yine sinir sistemi sayesinde vücudun ilgili organlarına gönderilir. Şekil 4.1'de görülen bir biyolojik sinir hücresi bilindiği gibi snapsler, soma, axon ve dentrite lerden oluşmaktadır. Sinir hücrelerindeki aksonlar miyelin olarak adlandırılan özel bir kılıfla kaplanmıştır. Miyelin kılıf, sinir iletimi sırasında oluşturulan elektriksel sinyallerin kaybını önlerken, uzak mesafelere hızlı bir şekilde iletilmelerini sağlar. Snapsler arasında bir fiziksel bağlantı olmayıp hücreler arası iletişim elektrik sinyallerinin bir hücreden diğerine geçmesi şeklinde olmaktadır. Somaya gelen sinyaller işlenerek sinir hücresi kendi elektrik sinyalini oluşturur ve axon ile dentrite lere gönderir. Sinir hücrelerinin birbirleri ile bilgi alışverişi snaptik bağlantılarda elektro-kimyasal yollarla olmaktadır. Canlı hücrelerinin içi ve dışı arasındaki iyon dağılımının dengesizliğinden dolayı hücrenin içindeki yük dışındaki yüke göre negatif yüklüdür[19].

Hücre içinde potasyum iyonları ile negatif yüklü iyonların bol bulunmasına karşın hücre dışında sodyum iyonlarının daha fazla bulunması sonucunda hücrelerin çoğunda yaklaşık -70 milivolt değerinde potansiyel enerji oluşmaktadır. Bu potansiyeller hücreler için büyük elektriksel dalgalardır. Söz konusu potansiyeller sinir hücrelerinin birbirleriyle haberleşmesini sağlayarak hücreler bilgi alışverişinde bulunur. Milyarlarca sinir hücresi bir araya gelerek sinir sistemini oluşturmaktadır. Yapay sinir ağları da sinir hücrelerinin bu özelliklerini esas alarak geliştirilmektedir[18][19][1].



Şekil 4.1 Biyolojik sinir hücresinin yapısı[20].

#### 4.1.2. Yapay sinir hücresi

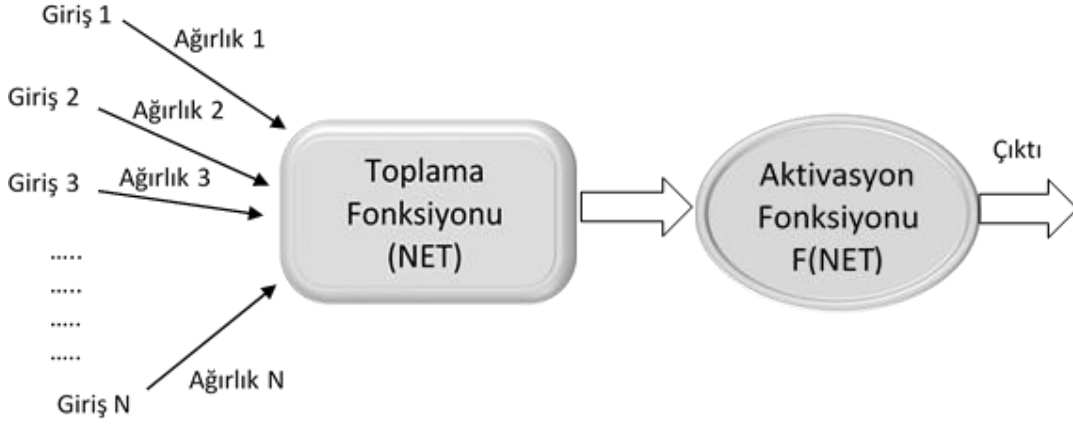
İnsan vücudundaki sinir ağlarını sinir hücreleri oluşturduğu gibi, yapay sinir ağlarını da yapay sinir hücreleri oluşturmaktadır. Yapay sinir hücreleri mühendislik literatüründe proses elemanları olarak da adlandırılmaktadır. Bir yapay sinir hücresi temel olarak 5 bölümden oluşmaktadır. Bunlar; girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkış olarak sıralanabilir[1]. Şekil 4.2’de yapay sinir hücresinin yapısı görülmektedir.

#### 4.1.3. Girişler

Yapay sinir ağı sistemine dış ortamdan veya başka bir hücreden gelen veriler girişleri oluştururlar. Geri beslemeli ağlarda ağıın ürettiği veri çıkışı aynı ağıın veri girişi olabilmektedir. Dış ortamdan gelen veriler ağı eğitmek için kullanılan verilerdir [18].

#### 4.1.4. Ağırlıklar

Ağırlıklar, hücreye ulaşan verilerin ağı üzerindeki etkisini belirleyen katsayı değerleridir. Ağırlık veri değerlerinin sayısal olarak büyük küçük olması önemli veya önemsiz olduğu anlamına gelmez. Yapay sinir ağının öğrenmesi, bu ağırlık değerlerinin belirlenmesi ile gerçekleşmektedir [1].



Şekil 4.2 Yapay sinir hücresinin yapısı

#### 4.1.5. Toplama fonksiyonu

Hücreye gelen net girdileri hesaplamak için kullanılan fonksiyondur. Genel olarak ağırlıklı toplam bulunur. Her giriş kendi ağırlığı ile çarpılarak toplanır. Sonuçta net girdi bulunmuş olur. Bu durumda ilgili matematiksel bağıntı denklem 4.1'deki gibi olmaktadır [18]

$$NetToplam = \sum_i^n x_i w_i \quad (4.1)$$

Denklem 4.1'de  $x$  girdileri,  $w$  ağırlıkları göstermektedir. Yapay sinir ağı modellenirken toplama fonksiyonu olarak mutlaka bu formülün kullanılması gerekmemektedir. Yapılacak işleme göre bazı durumlarda giriş verilerinin değeri dikkate alınırken başka bir durumda ise giriş veri sayısı önem arz etmektedir. Her problem için bulunmuş tek bir toplama fonksiyonu yoktur. Çoğu zaman deneme yanılma yöntemiyle probleme uygun toplama fonksiyonu belirlenmektedir. Toplama fonksiyonu olarak kullanılan değişik formüller 4.2'de gösterilmiştir[1].

Toplama fonksiyonuna birçok uygulamada  $\theta$  eşik değeri eklenerek, bütün girişlerin sıfır olması durumunda bile nöron çıkış değerinin sıfır olması engellenmektedir. Bu durumda toplama fonksiyonunu ifade eden bağıntı, denklem 4.2' deki gibi olur[18].

$$NetToplam = \sum_i^n x_i w_i + \theta \quad (4.2)$$

Denklemler 4.2’ de  $x$  girişleri,  $w$  ağırlıkları ve  $\theta$  eşik değerini ifade etmektedir. Eşik değeri giriş değerlerinden bağımsız olduğu için giriş değerleri sıfır olsa bile çıkış değerlerinin sıfır değil de eşik değerine eşitlendiği görülür.

Tablo 4.1 Toplama fonksiyonları ve açıklamaları

Net giriş	Açıklama
Çarpım $NetGirdi = \prod_i^n x_i w_i$	Giriş değerleri ile ağırlık değerleri çarpılır sonra elde edilen değerler birbirleri ile çarpılarak net girdi hesaplanır.
Maksimum (en çok) $NetGirdi = Max(x_i w_i), i = 1...N$	Giriş değerleri kendi ağırlıkları ile çarpıldıktan sonra elde edilen değerlerden en büyüğü(maksimum) net girdi olarak kabul edilir.
Maksimum (en az) $NetGirdi = Min(x_i w_i), i = 1...N$	Giriş değerleri kendi ağırlıkları ile çarpıldıktan sonra elde edilen değerlerden en küçüğü(minimum) net girdi olarak kabul edilir.
Çoğunluk $NetGirdi = \sum_{i=1}^n sgn(x_i w_i)$	Giriş değerleri kendi ağırlıkları ile çarpıldıktan sonra negatif olanlar ile pozitif olanların sayısı bulunur. Bulunan sayılardan büyük olan hücrenin net girdisi olarak kabul edilir.
Kümülatif Toplam $NetGirdi = Net(eski) + \sum_{i=1}^n x_i w_i$	Gelen bilgiler ağırlıklı olarak toplanır ve daha önce gelen bilgilere eklenerek hücrenin net girdisi elde edilir.

#### 4.1.6. Aktivasyon fonksiyonu

Toplama fonksiyonu sonucu elde edilen net veriyi işleyerek hücrenin bu veri girişine karşılık üretilecek olan çıkış verisini belirler. Aktivasyon fonksiyonu genellikle doğrusal olmayan fonksiyonlardan seçilir. Aktivasyon fonksiyonunun doğru seçilmesi ağırlık performansını ve doğru çalışmasını doğrudan etkileyecektir. Aktivasyon fonksiyonunun kullanım amacı, toplama fonksiyonundan elde edilen sonuca göre çıktı değerlerine dönüştürmek. Genellikle aktivasyon fonksiyonları net girdiyi +1 ile -1 veya 0 ile 1 arasında eşitler.

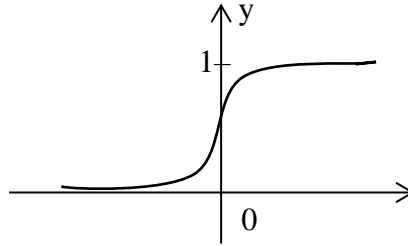
Özellikle sınıflamanın yapılacağı durumlarda bu tür aktivasyon fonksiyonları kullanılabilir. Lineer aktivasyon fonksiyonları ise, reel bir sonuç elde edilmek istendiğinde kullanılan aktivasyon fonksiyonlarıdır.

Toplama fonksiyonunu elde etmek için farklı fonksiyonlar kullanılabildiği gibi, aktivasyon değerini hesaplamak içinde farklı fonksiyonlar kullanılabilir. Uygulamada en çok tercih edilen aktivasyon fonksiyonları aşağıda maddeler halinde sıralanmış ve açıklanmıştır [1]:

#### 4.1.6.1. Sigmoid fonksiyonu

Aktivasyon fonksiyonu olarak özellikle çok katmanlı ağ modellemelerinde ve birçok uygulamada oldukça popüler olan sigmoid fonksiyonu tercih edilmektedir. Sigmoid fonksiyonunun formülü denklem 4.3' de, şekli ise şekil 4.3' de gösterilmiştir[21].

$$y = f(NET) = \frac{1}{1 + e^{-NET}} \quad (4.3)$$

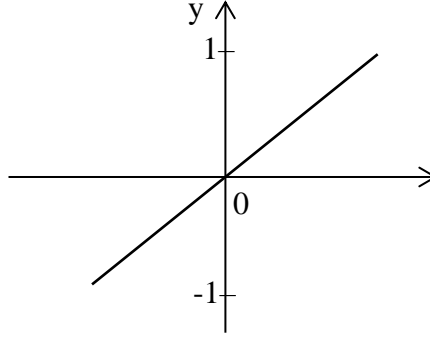


Şekil 4.3 Sigmoid fonksiyonu

#### 4.1.6.2. Lineer (Doğrusal) fonksiyon

Hücreye gelen verileri olduğu gibi çıkışa veren fonksiyondur. Daha çok ADALINE (Adaptive Linear Element) ağlarında kullanılır. Lineer (Doğrusal) fonksiyonun formülü denklem 4.4' te, şekli ise Şekil 4.4'te gösterilmiştir.

$$y = NetToplam = \sum_i^n x_i w_i + \theta \quad (4.4)$$

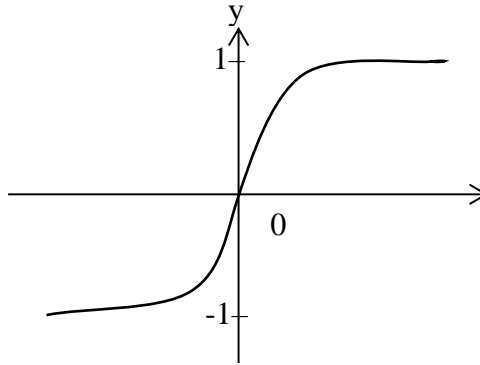


Şekil 4.4 Lineer fonksiyonu

#### 4.1.6.3. Hiperbolik tanjant fonksiyonu

Uygulamada yaygın olarak kullanılan bir diğer aktivasyon fonksiyonu olan hiperbolik tanjant fonksiyonu gelen NET girdi değerini tanjant fonksiyonuyla hesaplamasından elde edilen değerdir. Sigmoidal fonksiyonunun nonlineeritenin yaygın olarak kullanılan başka bir şekli genel haliyle tanımlandığı hiperbolik tanjant fonksiyonudur. Fonksiyonun formülü denklem 4.5'te, şekli ise Şekil 4.5' te gösterilmiştir.

$$y = F(NE T) = \frac{e^{NE T} + e^{-NE T}}{e^{NE T} - e^{-NE T}}$$



Şekil 4.5 Hiperbolik tanjant fonksiyonu

(4.5)

#### 4.1.6.4. Step fonksiyonu

NET girdi değerinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır. Fonksiyonun formülü denklem 4.6'da görülmektedir.

$$F(Net) = \begin{cases} 1 & \text{eğer } NET > \text{ eşikdeğer} \\ 0 & \text{eğer } NET \leq \text{ eşikdeğer} \end{cases} \quad (4.6)$$

#### 4.1.6.5. Sinus fonksiyonu

Öğrenilmesi istenilen olaylar sinüs fonksiyonuna uygun dağılım gösteriyorsa sinüs fonksiyonu kullanılır. Fonksiyonun formülü denklem 4.7'de görülmektedir.

$$F(NET) = \text{Sin}(NET) \quad (4.7)$$

#### 4.1.6.6. Eşik değer fonksiyonu

Gelen bilgilerin 0 ya da 1'den büyük veya küçük olmasına göre bir değer alır. 0 ve 1 arasında değerler alabilir haricinde değer alamazlar. Fonksiyonun formülü denklem 4.8'de görülmektedir.

$$F(NET) = \begin{cases} 0 & \text{eğer } NET \leq 0 \\ NET & \text{eğer } 0 < NET < 1 \\ 1 & \text{eğer } NET \geq 1 \end{cases} \quad (4.8)$$

#### 4.1.7. Hücrenin çıktısı

Aktivasyon fonksiyonunca belirlenen çıktı değerinin sistem dışına veya diğer nöronlara iletilmesidir. Elde edilen çıkış değerini hücre kendisine tekrar giriş değeri olarak ta gönderebilir. Bir işlem elemanının birçok giriş değeri olabilirken tek bir çıktı değeri olmaktadır.

Yapay sinir ağının girişine verilen verileri işleyerek anlamlı çıkışlar verebilmesi için ağın eğitilmesi gerekmektedir. Yapay sinir ağlarında yapay sinir hücrelerinin bağlantılılarının ağırlık değerlerinin belirlenmesi işlemine ağın eğitilmesi denir. Ağırlık değerleri ilk olarak rastgele seçilir. Yapay sinir ağının öğrenmesi için örnekler gösterildikçe bu ağırlık değerleri değişir. Buradaki amaç yapay sinir ağına gösterilen örnekler için doğru çıktıları verecek ağırlık değerlerini bulmaya çalışmaktır. Doğru ağırlık değerini bulmaya çalışmak için örnekler ağa birçok kere gösterilir[1].

Örneklerin temsil ettiği olaylar hakkında, yapay sinir ağı genellemeler yapabilme yeteneğini edinmişse, doğru ağırlık değerlerine ulaşılmış demektir. Bu genelleştirme yeteneğin kazanması işlemine ağın öğrenmesi denir. Genel olarak yapay sinir ağları ağın yapısına göre ileri beslemeli ve geri beslemeli olarak ikiye, öğrenme stratejisine göre danışmanlı, danışmansız ve destekleyicili olarak üçe ayrılır [1].

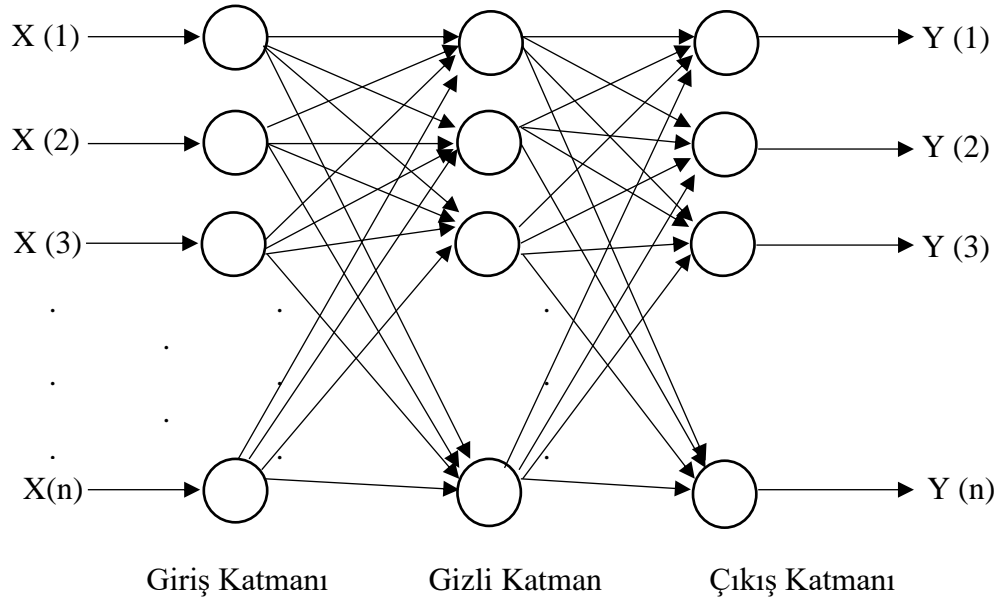


## 4.2. İleri Beslemeli Yapay Sinir Ağı

Bu modeldeki ağlarda bir katmandaki tüm nöronların çıkışları bir sonraki katmandaki nöronların girişleridir. Girdilerin uygulandığı katmana giriş katmanı, çıkışların alındığı katmana çıkış katmanı denir. Giriş ve çıkış katmanlarına dış ortamdan yani gerçek dünyadan erişilebilir. Giriş ve çıkış katmanları arasında gizli katmanlar bulunur.

Bu katmanlara giriş katmanındaki nöronlardan bilgiler ulaşır ve bu katmanlarda bilgiler işlenir. Bir gizli(ara) katman ile birçok probleme çözüm üretilebilir. Fakat istenilen sonuç alınamazsa birden fazla ara katman kullanılabilir.

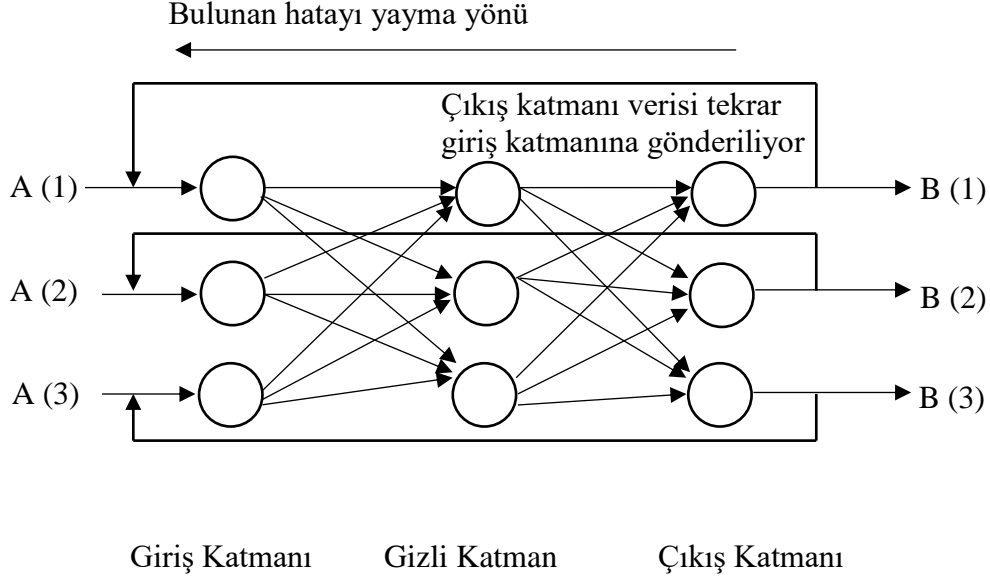
Çıkış katmanı ara katmandan gelen bilgileri işleyerek dış dünyaya veri çıkışını iletir. Bu ağların çıkış verilerinin geriye, giriş katmanına veya diğer katmanlara geri veri iletimi yoktur[18]. Şekil 4.6'da ileri beslemeli yapay sinir ağı modeli gösterilmiştir.



Şekil 4.6 İleri beslemeli yapay sinir ağı modeli

## 4.3. Geri Beslemeli Yapay Sinir Ağı

Nöronların çıktıları sadece ileri doğru değil, geri doğru da gönderilebilmektedir. Bu tür ağ yapısında bir nöronun çıktı değeri kendisine, başka bir nörona, ağa belirli bir şekilde geri döndürülerek giriş verisi olabilmektedir. Geri besleme bir katmandaki hücreler arasında da olabilmektedir[21]. Şekil 4.7'de geri beslemeli bir yapay sinir ağı modeli örneği görülmektedir.



Şekil 4.7 Geri beslemeli yapay sinir ağı modeli

#### 4.4. K-Means

Danışmansız öğrenme yöntemlerinden biri olan k-means, eşit büyüklükteki küresel kümeleri bulmayı hedefler. Giriş parametresine  $x$  denirse, k-means algoritması  $m$  tane nesneyi  $n$  tane kümeye bölerek öncelikle giriş parametresi olarak  $x$  değerinin verilmesini gerektirir. Küme içi benzerliğinin yüksek olması amaçlanırken, kümeler arası benzerliğin düşük olması asıl amaçtır. Küme benzerliğinin ölçümü ise kümedeki objelerin ortalama değeri ile ölçülür, bu da kümenin ağırlık merkezi olur.[18].

K-means algoritmasına göre ilk olarak her biri bir kümenin merkezini veya ortalamasını temsil etmek üzere  $m$  tane nesne seçilir. Kalan nesnelere, kümelerin ortalama değerlerine olan uzaklıkları göz önüne alınarak en benzer oldukları kümelere dahil edilir. Sonra ortalama değeri hesaplanan her bir kümenin yeni küme merkezleri belirlenir ve tekrar nesne-merkez uzaklıkları incelenir[21].

K-means kümeleme yönteminin değerlendirilmesinde sıklıkla karesel hata kriteri SSE kullanılır. En iyi sonuca ulaşmak için en düşük SSE değerine sahip kümelemeyi elde etmek gerekmektedir. SSE ifadesi aşağıda verilmiştir:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x) \quad (4.9)$$

Denklem 4.9'da  $x$ ,  $C_i$  kümesinde bulunan bir nesne,  $m_i$   $C_i$  kümesinin merkez noktası. Bu kriterlerin sonucunda,  $k$  tane kümenin mümkün olduğunca yoğun ve birbirinden ayrı olarak sonuçlanması amaçlanır. K-means algoritması karesel hata fonksiyon değerini azaltacak  $k$  değerini tespit etmeye çalışır[21].

K-means algoritmasının da birçok algoritmalarda olduğu gibi avantajları ve dezavantajları bulunmaktadır. Avantajları arasında, uygulanabilirliğinin kolay olması büyük veri kümelerinde hızlı çalışabilir olması, büyük veri setlerini işlerken nispeten ölçeklenebilmesi gösterilebilir. K-means algoritması, kümelerin birbirinden ayrılma işlemi iyi yapılmışsa ve yoğun noktalardan oluşuyorsa iyi sonuçlar üretebilmektedir.

Diğer taraftan k-means algoritmasının birçok zayıf yönü bulunmaktadır. Başlangıçta küme sayısının tanımlanmasını gerekmesi, kullanıcının başlangıçta  $k$  sayısını belirleme zorunluluğu, kategorize edilmiş veriler gibi bazı uygulamalarda başarısız olması temel dezavantajlarıdır. Küresel olmayan, yoğunlukları farklı olan ve farklı büyüklüklerden oluşan kümeleri içeren veri kümelerini bulmada uygun bir yöntem değildir[18].

#### 4.5.FCM (Fuzzy c-means)

FCM algoritması, nesnelere birden fazla kümeyle ait olmasına izin verir ve amaç fonksiyonu temelli bir metottur. Fuzzy(bulanık) mantık prensibince her bir veri, kümelerin her birine 0-1 arasında değişen bir üyelik değeri ile bağlıdır. Bir verinin tüm sınıflara ait üyelik değeri toplamı 1 olmalıdır. Nesne hangi küme merkezine yakın ise, yakın olduğu kümeyle ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olacaktır. Amaç fonksiyonunun hedeflenen minimum ilerleme değerine yaklaşmasıyla kümeleme işlemi sonuçlanır. K-means algoritması, denklem 4.10'daki en küçük kareler yönteminde amaç fonksiyonunu öteleyerek minimize etmek için çalışır [22].

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty \quad (4.10)$$

İlk olarak  $U$  üyelik matrisi rastgele seçilerek algoritma başlatılır daha sonra merkez vektörleri hesaplanır. Merkez vektörleri aşağıdaki ifadeye bağlı olarak hesaplanır[22]:

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N u_{ij}^m} \quad (4.11)$$

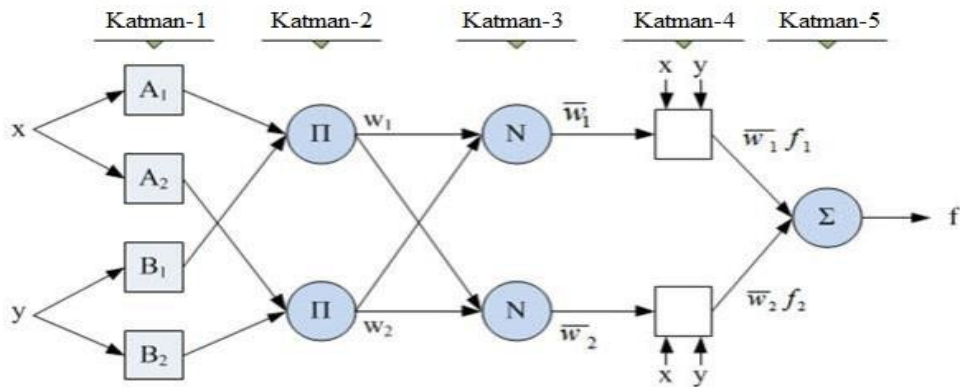
Küme merkezleri hesaplandıktan sonra, U matrisi denklem 4.12'deki eşitlik kullanılarak tekrar hesaplanır. Önceki U matrisi ile yeniden hesaplanan U matrisi karşılaştırılarak aradaki fark kontrol edilir. Fark  $\varepsilon$ 'dan küçük değere ulaşana kadar işlemler devam eder[23].

$$U_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{|x_i - c_k|}{|x_i - c_j|} \right)^{\frac{2}{m-1}}} \quad (4.12)$$

İşlemler sonucunda fuzzy değerler içeren U üyelik matrisi kümeleme işleminin sonucunu verir. İstenildiği takdirde iyileştirme işlemleri yapılarak elde edilen değerler yuvarlanarak 0 ve 1'lere dönüştürülebilir[23].

#### 4.6. ANFIS (Adaptive Neuro Fuzzy Inference System)

Bulanık ara yüzülü yapay sinir ağları (ANFIS), yapay sinir ağları ve bulanık mantık sistemlerinin özelliklerini taşıyan hibrid bir sistemdir. Bulanık mantıkta matematiksel değişkenler yerine insanın karar vermesinde kullandığı mekanizmayı taklit ederek, kelimelerle ifade edilebilen değişkenler kullanılır. Alışlagelen mantıkta bir değişkenin alabileceği değer 1 veya 0 olarak belirtilir. Fakat bulanık mantıkta söz konusu değer 0 ile 1 arasında bir değer olabilir. Bulanık mantığın sadece 0 ve 1 leri işlemeyip belirsiz verileri de işleyebilmesi yeteneğinden ve yapay sinir ağının öğrenme özelliğinden yararlanabilmek için bu iki sistem değişik yöntemler kullanarak bir araya getirilip birleştirilmektedir[24]



Şekil 4.8 İki girişli anfis mimarisi

Beş katmandan oluşan Şekil 4.8’de gösterilen anfis mimarisinde birinci katmanda,  $A_i$  ve  $B_i$  değerleri bir üyelik fonksiyonuyla üyelik değerlerini alırlar. Üyelik fonksiyonu denklem 4.13’deki gibi bir Gauss üyelik fonksiyonu ve denklem 4.14’deki gibi Bell aktivasyon fonksiyonu kullanılmaktadır[25].

$$\mu_{A_i}(x) = \exp \left[ - \left( \frac{x - c_i}{2a_i} \right)^2 \right] \quad (4.13)$$

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}} \quad (4.14)$$

İkinci katmanda her düğüm sabit ve daire düğümü  $\pi$  olarak etiketlenir. Çıkış düğümü, düğüme gelen verilerin çarpılması sonucu elde edilen verinin bir sonraki düğüme iletilmesidir. Bu katmandaki her düğüm her kural için tetikleme gücünü temsil eder. İkinci katmanda her bir kuralın tetikleme gücünü temsil eden çıktıyı elde etmek için denklem 4.15’deki fonksiyon kullanılır[25].

$$M_{2i} = w_i = \mu_{A_i}(x) * \mu_{B_i}(y), \quad i = 1, 2 \quad (4.15)$$

Üçüncü katman, normalizasyon katmanı olarak ta bilinen bu katmanda tetikleme gücü değerleri toplam ateşleme gücü değerlerine oranlanarak normalize edilir. Söz konusu işlem denklem 4.16’da gösterilmiştir[25].

$$M_{3i} = \bar{w}_i = \frac{w_i}{\sum_i w_i} \quad (4.16)$$

Dördüncü katman, berraklaştırma katmanı olarak ta bilinir. Çıktı değerlerini y kural tabanından çıkarır. Çıkış değişkeni sabit bir sayı olabileceği gibi değişkene bağlı polinom bir fonksiyonda olabilir. Bu katmandaki düğümlerin çıkış değeri denklem 4.17’deki gibi olmaktadır. Denklemdaki ( $p_i$ ,  $q_i$ ,  $r_i$ ) kural sonucu parametre setidir[24].

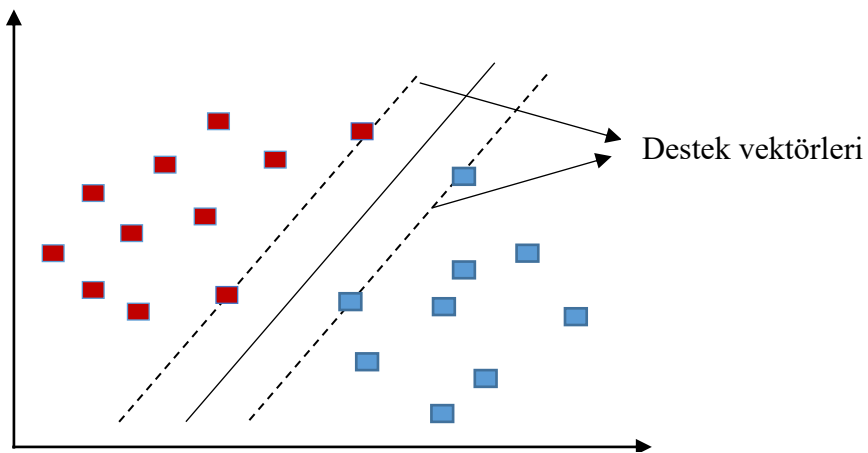
$$M_{4i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (4.17)$$

Beşinci katman, çıktı katmanı olarak ta bilinir. Sadece bir düğümü vardır ve denklem 4.18’de görüldüğü gibi ANFIS çıkış değerini elde etmek için bulanıklaştırma katmanının düğümlerinden gelen tüm değerlerin toplamını hesaplar[24].

$$M_{5i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (4.18)$$

#### 4.7. Destek Vektör Makinesi (SVM - Suport Vector Machine)

SVM sınıflandırma problemlerinin çözümünde, regresyon işlemlerinde, örüntü tanımada kullanılan makine öğrenme algoritmasıdır. SVM’de asıl amaç sınıfları birbirinden ayırmak için bir düzlem elde etmektir. Sınıfları birbirinden ayırmak için çok fazla düzlem olacağından, sınıfları ayıran düzlemin her iki sınıfa da en uzakta olacak şekilde ayarlanabilmesidir[26]. Söz konusu düzleme de en yakın olan vektörler de destek vektörleri olarak adlandırılır. Şekil 4.9’da örnek destek vektör makinesi görülmektedir. Eğitim sonucu elde edilen verilerden oluşan destek vektörleri sınıflandırma işlemlerinde kullanılmaktadır. SVM, veri setinin doğrusal olarak ayrılma veya doğrusal olarak ayrılmama durumuna göre genel olarak iki bölüme ayrılmaktadır. Bunlar doğrusal destek vektör makineleri ve doğrusal olmayan destek vektör makineleridir [26].



Şekil 4.9 Destek vektör makinesi

#### 4.7.1. Doğrusal destek vektör makinesi

Orijinal olarak ikili sınıflandırma için tasarlanan karar sınırları doğrusal bir fonksiyon ile temsil edilir. Hiper düzlem üzerindeki L düzlemi denklem 4.19'daki gibi ifade edilebilir[26].

$$L = w^T x + b = 0 \quad (4.19)$$

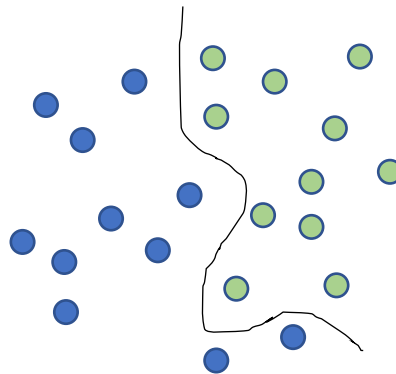
Denklem 4.19 denklem 4.20 gibi de yazılabilir.

$$\sum_{i=0}^m w_i x_i + b = 0 \quad (4.20)$$

Denklem 4.20'de  $W$  ağırlık vektörünü  $W=(W_1, W_2, \dots, W_m)$ ,  $m$  ise sayısını belirtmektedir. Denklemde bulunan  $b$  ise bir sabit sayıyı ifade etmektedir.

#### 4.7.2. Doğrusal olmayan destek vektör makinesi

Veri setinin doğrusal olarak tam veya belirli bir hata ile ayıramadığı durumlarda doğrusal destek vektör makinesi yerine doğrusal olmayan destek vektör makinesi kullanılır. Çözümü istenen problemin verilerinin doğrusal ayrılama durumu şekil 4.10'da gösterilmiştir. Doğrusal olmayan problemlerin çözümünde kernel(çekirdek) fonksiyonları ile çözümü istenen problemin verileri ilk olarak daha yüksek boyutlu ve doğrusal olarak ayrılabilirleri bir alana taşınır. Daha sonra, oluşturulan bu yeni alanda çözüme ulaşılmaya çalışılır[27]



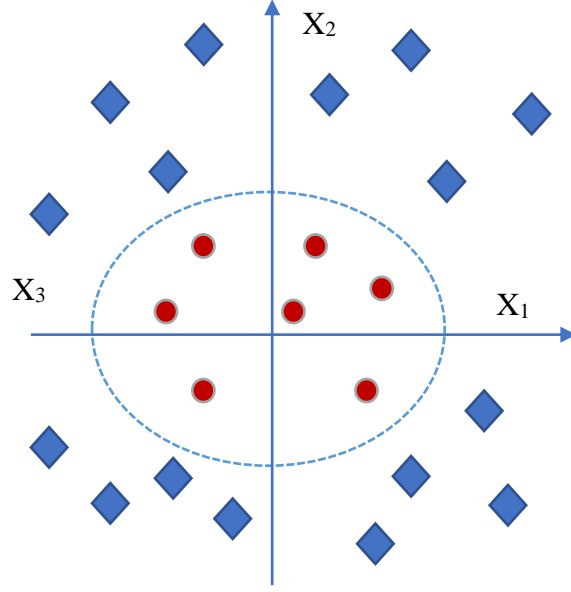
Şekil 4.10 Doğrusal ayrılama durumu

Elde edilen iki boyutlu verilerin doğrusal olarak ayıramadığı durumlarda veri setinin üç boyutlu uzaya taşınarak verilerin doğrusal olarak ayrımı yapılır. Bu durum Şekil 4.11' a) ve Şekil 4.11' b) de gösterilmiştir. Doğrusal olmayan destek vektör makineleri için hiper düzlemine bağlı olarak sınıflandırıcı fonksiyonu, özellik uzayında tanımlı olmak kaydıyla denklem 4.21'deki gibi ifade edilmektedir [27].

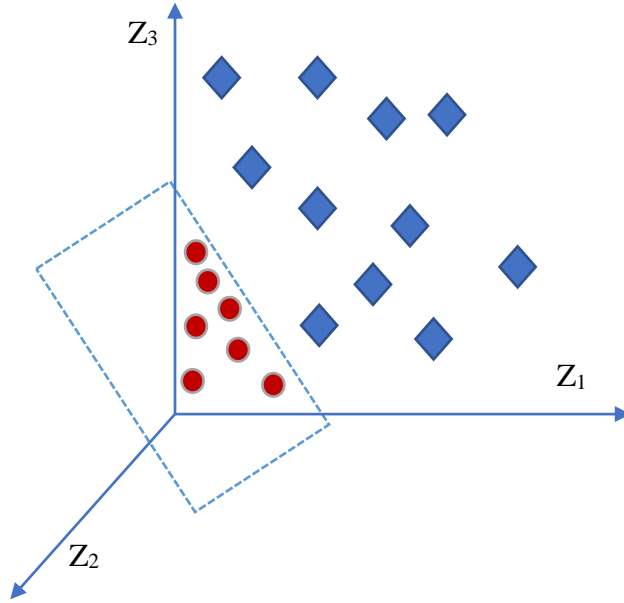
$$f(x) = \text{sgn} \left( \sum_{i=1}^n a_i y_i K(x, x_i) + b \right) \quad (4.21)$$

Denklem 4.21’de  $K(x, x_i) = \Theta^T(x) \Theta_i(x)$  Çekirdek fonksiyonu olup simetriktir.  $K(x, x_i) = K(x_i, x)$  olarak ifade edilmektedir. Destek vektör makinesi, yapısal risk minimizasyonu prensibine dayanarak yüksek boyutlu verileri minimum sorunla çözmüştür. SVM, metin kategorizasyonu, zaman dizisi tahmini, örüntü tanıma ve görüntü işleme gibi birçok alanda başarılar elde etmesine rağmen yine de derinlemesine araştırıldığında birçok eksikliğe sahip olduğu görülmüştür. Örneğin istatistiksel öğrenme teorisi ile diğer teorik sistem arasındaki ilişkilerde, büyük verilerin işlenmesinde, parametrelerin seçimi, belirli bir problemin genelleme yeteneğini ve uygulamaların sınırlamalarını içeren problemlerin çözümünde sorunlar bulunmaktadır[27].





Şekil 4.11 a) Doğrusal olmayan veri setinin 3 boyutlu uzaya taşınarak sınıflandırılması

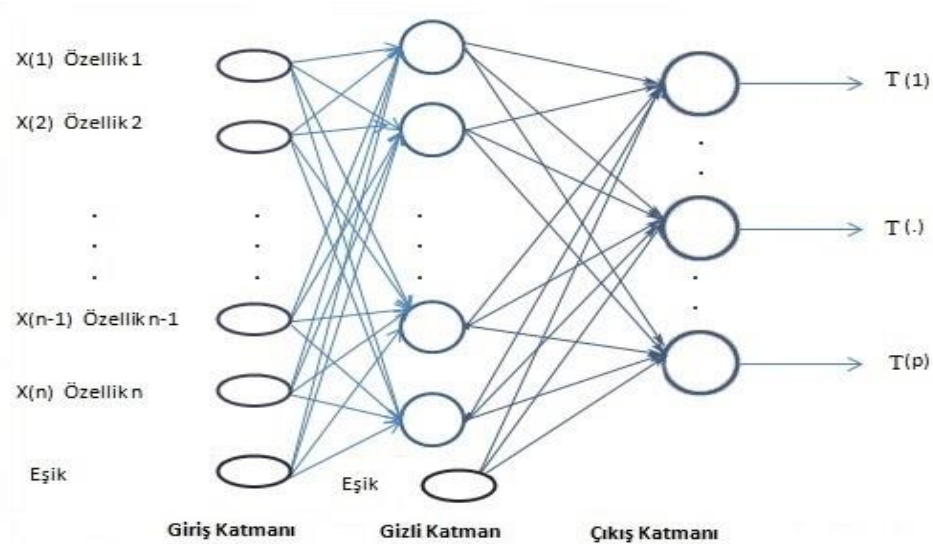


Şekil 4.11 b) Doğrusal olmayan veri setinin 3 boyutlu uzaya taşınarak sınıflandırılması

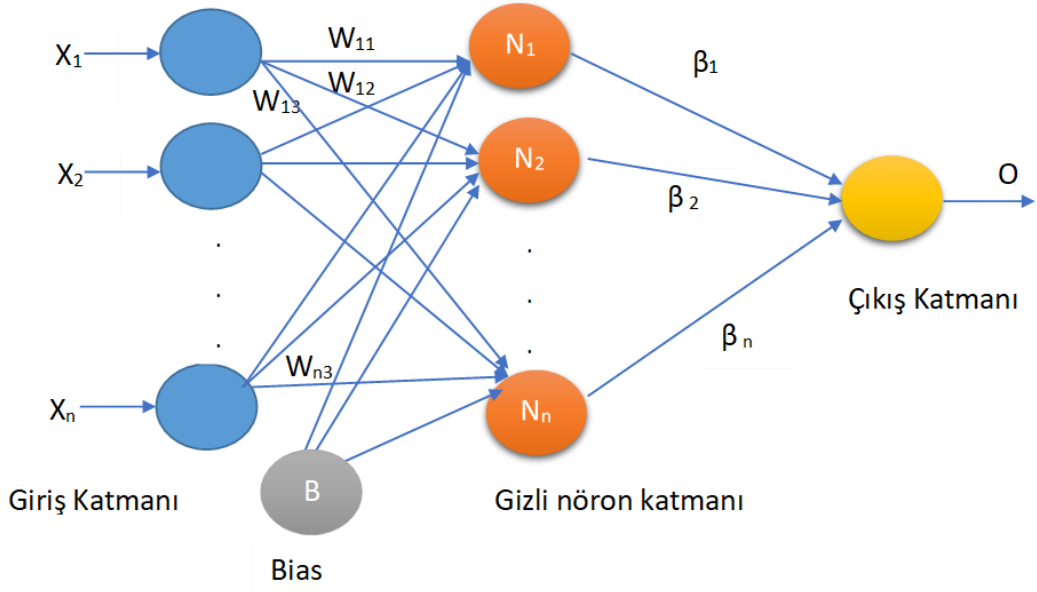
#### 4.8. Aşırı Öğrenme Makinesi (ELM - Extreme Learning Machine )

İleri beslemeli yapay sinir ağları, birçok alanda yeteneklerinden dolayı kullanılmaktadır. Klasik parametre teknikleri kullanarak ele alınması zor olan geniş bir doğal ve yapay olgu sınıfına model sağlamasının yanında daha hızlı bir öğrenme altyapısı sunmaktadır. Geleneksel öğrenme algoritmaları çoğunlukla gerekenden daha yavaştır. Geleneksel yöntemler kullanılarak yapay sinir ağlarını eğitmek birkaç saat, birkaç gün hatta daha fazla zaman alabileceğini görmek olasıdır. Huang ve arkadaşları yapay sinir ağlarını eğitmek için ELM' yi önermişlerdir[2]. ELM giriş ağırlıkları rastgele çıkış ağırlıkları ise analitik olarak hesaplanan tek gizli katmanlı ileri beslemeli bir yapay sinir ağı modelidir. ELM de gizli katmanda sigmodial, sine, Guassian ve hard-limitin gibi aktivasyon fonksiyonları kullanılırken çıkış katmanında doğrusal fonksiyon kullanılmaktadır. ELM de türevlenemeyen veya kesikli aktivasyon fonksiyonları kullanılabilir.

Geleneksel ileri beslemeli yapay sinir ağlarında gradyan bazlı öğrenme algoritmaları ile ağırlıklar ve eşik değerleri gibi parametrelerin güncellenmesi gerekir. Bu sistemlerde öğrenme süreci zaman almakta hem de hata yerel bir noktaya takılabilmektedir. Momentum değeri değiştirilerek hatanın bir noktaya takılmasını engellemek mümkün olsa da bu durum öğrenme işlemi sürecini kısaltmayacaktır. ELM tek gizli katmanlı ileri beslemeli bir ağ olduğu için giriş çıkış ağırlıkları ve eşik değerleri ağın performansını olumsuz yönde etkilememektedir. Şekil 4.12'de tek gizli katmana sahip ileri beslemeli bir yapay sinir ağı modeli görülmektedir.



Şekil 4.12 Tek gizli katmana sahip ileri beslemeli bir YSA



4.13 Tek gizli katmana sahip ELM modeli

Bu modelde giriş ağırlıkları rastgele çıkış ağırlıkları ise analitik olarak hesaplandığından, ELM benzer diğer algoritmalara göre bazı görevlerde daha iyi performans sunarken daha da hızlı olabilmektedir [2]. Geleneksel öğrenme algoritmalarından farklı olarak, sadece en küçük eğitim hatasını değil aynı zamanda en küçük ağırlık seviyesine ulaşma eğiliminde olması daha iyi bir genelleme performansı sunmasını sağlamaktadır[2][3]. Şekil 4.13’de  $X = (X_1, X_2, X_3, \dots, X_n)$  giriş ve  $O$  çıkış özelliklerini belirtmek üzere, gizli katmanda  $S$  nörona sahip ağırlık matrisi aşağıdaki gibi belirtilir[2].

$$\sum_{i=1}^S \beta_i g(W_i X_j + b_i) = o_j, \quad j = 1, 2, \dots, S \quad (4.22)$$

Denklem 4.22’de:

$W_i = (W_{i1}, W_{i2}, \dots, W_{in})$  giriş katmanındaki ağırlıkları ifade eder.  $\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})$  çıkış katmanındaki ağırlıkları belirtir.  $\beta = H^+ T$  ifadesi hesaplanarak  $\beta$  elde edilir.  $H^+$ ,  $H$  çıkış matrisinin genelleştirilmiş tersi Moore-Penrose matrisi olarak tanımlanmıştır.  $T$  hedef özelliklerdir.  $b_i$  gizli katman nöronların eşik değerleridir.

$O_j$  çıkış değerleridir,  $g(x)$  aktivasyon fonksiyonudur[2].  $N$  girişli bir ağda amaç hatası  $\sum_{j=1}^N ||o_j - t_j|| = 0$  olması hedeflenmiştir. Bu nedenle denklem 4.22, denklem 4.23 gibi yazılabilir.

$$\sum_{i=1}^S \beta_i g(W_i X_j + b_i) = t_j, \quad j=1, 2, \dots, N \quad (4.23)$$

Denklem 4.23 eşitliğinde  $H\beta=T$  yazılabilir. Bu eşitlikteki  $H$  denklem 4.24'deki gibi,  $\beta$  ve  $T$  ise denklem 4.25'de ki gibi ifade edilir.

$$H(w_1, \dots, w_S, b_1, \dots, b_N, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 x_1 + b_1) & \cdots & g(w_S x_1 + b_S) \\ \vdots & \cdots & \vdots \\ g(w_1 x_N + b_1) & \cdots & g(w_S x_N + b_S) \end{bmatrix}_{N \times S} \quad (4.24)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_S^T \end{bmatrix}_{S \times m} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (4.25)$$

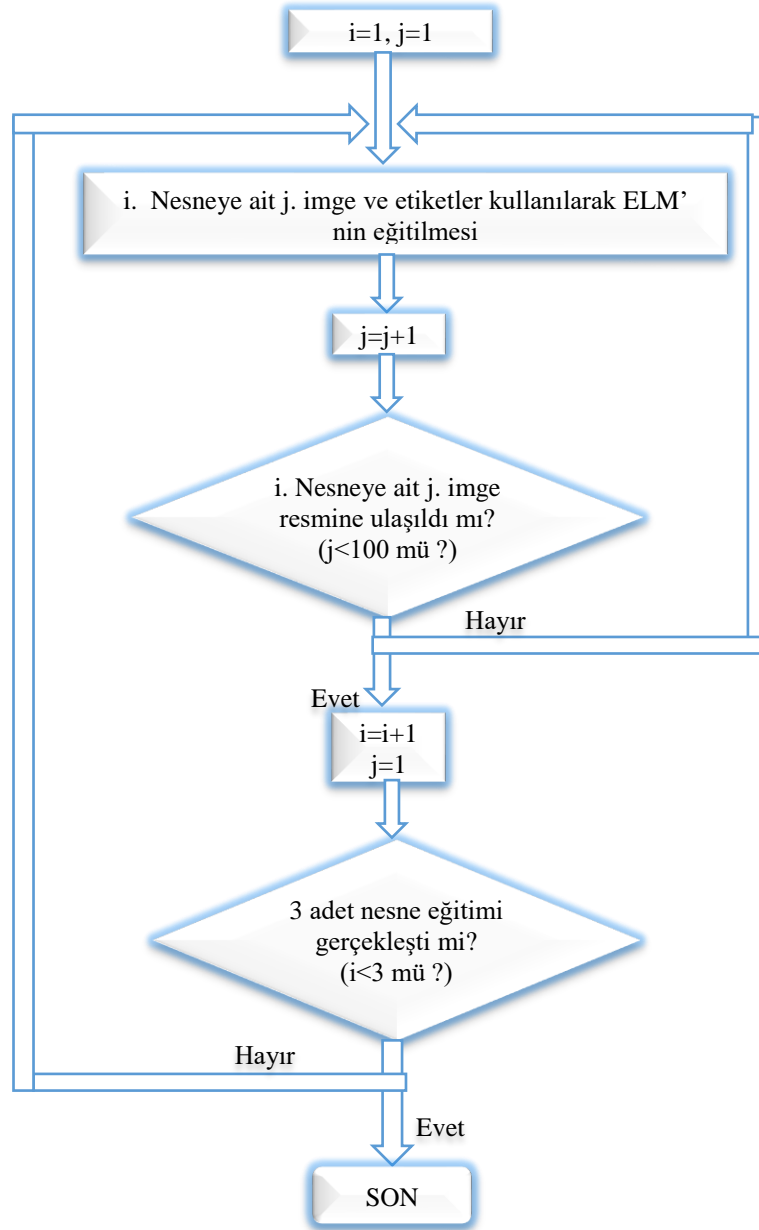
Denklem 4.24'de  $H$  gizli katman çıkış matrisidir. ELM'de giriş ağırlıkları  $W_i = (W_{i1}, W_{i2} \dots W_{in})$  ve  $b_i$  gizli katman nöronlarının eşik değerleri rastgele üretilirken, gizli katman çıkış matrisi analitik olarak hesaplanmaktadır. Geleneksel ileri beslemeli bir yapay sinir ağı modelinin eğitilmesi için, ELM algoritması kullanılarak  $H\beta=T$  doğrusal denklemde en küçük kareler çözümünün aranması işlemi gerçekleştirilmektedir[2].

### 5.3D SEGMENTASYON İŞLEMİ İÇİN ÖNERİLEN YÖNTEM

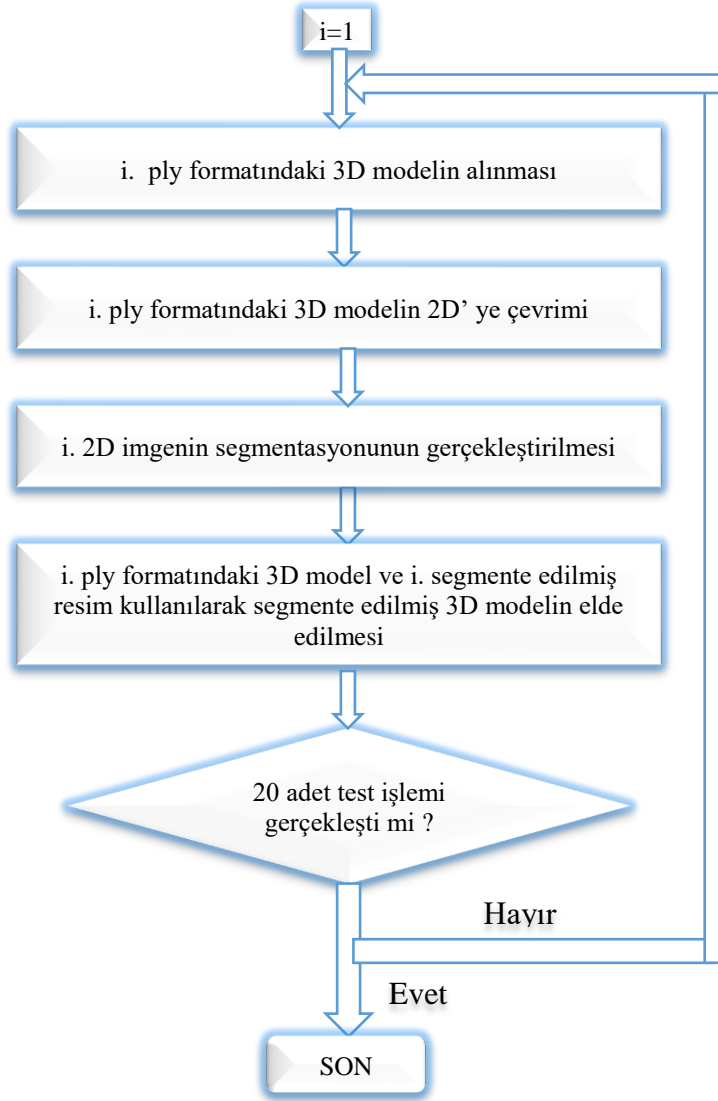
Bu çalışmada segmentasyon işlemi için Aşırı Öğrenme Makinesi (ELM) kullanılmıştır. ELM ye ait önemli özelliklerinden bazıları, genişletilebilir bir performans sunarken aynı zamanda hızlı olması, klasik yapay sinir ağlarındaki gibi öğrenme oranı için bir parametreye ihtiyaç duymaması, yapılacak olan işleme göre değişik aktivasyon fonksiyonlarının da kullanılabilmesi, eğitim sürecinin kısa sürmesi söylenebilir.

3D görüntüler üzerinde direkt segmentasyon işlemi yapıldığında segmentasyon süresi çok uzamaktadır. Ayrıca kompleks 3D modeller üzerinde de doğrudan segmentasyon yapıldığında birçok zorlukların yaşanması ve gürültülerin oluşması gibi sorunlar göze çarpmaktadır. Geliştirilen metotla yalnızca istenen nesnenin 3D modellenmesi gerçekleştirilmiş olacaktır. Böylece bu yöntemin ve ELM'nin avantajını da kullanarak daha yüksek işlem hızıyla ve daha gerçekçi segmentasyon sonucuna ulaşılmaktadır.

Önerilen yönteme ait diyagram şekil 5.1'de gösterilmiştir. Birinci aşama 3D segmentasyon işlemi sonucu elde edilecek nesnelerin 2D görüntülerinin, ELM eğitim seti için ön işlemlerden geçirilerek hazırlanması. İkinci aşama, hazırlanan görüntülerin ELM eğitim seti bölümüne iletilerek ELM'nin eğitilmesi. Üçüncü aşama, segmentasyonu yapılacak 3D görüntünün 2D dönüştürülmesi için gerekli işlemlerden geçirildikten sonra elde edilen görüntü ELM'nin test data bölümüne gönderilmesi. Dördüncü aşama, ELM'nin sonucuna göre orijinal 3D görüntüden segmentasyon işlemi sonucu istenilen nesne veya nesnelerin elde edilmesidir.



Şekil 5.1 Önerilen yöntemin eğitim sürecine ait akış diyagramı



Şekil 5.2 Önerilen yöntemin test sürecine ait akış diyagramı

### 5.1. ELM'nin Eğitimi

ELM'nin eğitiminde aktivasyon fonksiyonu olarak sigmoid seçilmiştir. Eğitim verilerine bağlı olarak gizli katmandaki gizli düğüm parametreleri oluşturulup, gizli katman çıkışları hesaplandıktan sonra çıkış ağırlık vektörleri hesaplanır. ELM'nin eğitim sürecinin verildiği Algoritma 5.1' den görüldüğü gibi ilk olarak eğitimi yapılacak objenin görüntüleri, görüntü işleme teknikleri kullanılarak ön işleminden geçirilir. Elde edilen görüntüler daha sonra ELM'nin training(öğrenme) data bölümüne gönderilerek ELM'nin datasetine eklenir. İlgili objeye ait etiket verisi de ELM'nin label data bölümüne gönderilir. Ardından 20 adet görüntüye ait data ve etiket bilgileri kullanılarak ELM'nin eğitimi gerçekleştirilmektedir.

### Algoritma 5.1 ELM'nin eğitimi

1. Etiket (3x100) = ELM' nin eğitiminde kullanılacak etiket bilgilerinin girilmesi
2.  $V=10304$  (92 x 112) x 3 ölçülerinde matris oluştur
3. For j=1: 3
4. For i=1: 100
5. Resim=j. nesne eğitimi için i. resmi oku
6. Resim= resim imgesini 10304 (92 x 112) x1 boyutuna çevir
7.  $Y$ = resim imgesini ön işlemde geçir
8. Resim imgesini  $V(:, i)$  matrisine görüntüyü binary formatta kaydet( $Y$ )
9. End End
10.  $V$  ve Etiket matrislerini kullanarak ELM' nin eğitilmesi

Algoritma 5.1'in adımları:

1. Eğitimi yapılacak objelerin isimlerinin ilgili matrise girilmesi
2. Her bir obje için 10304 boyutunda vektör oluşturuluyor. ELM'nin eğitiminde kullanılacak her bir imgenin boyutu 92 x 112 olduğundan 10304 boyutunda vektör oluşturulmaktadır. İmge boyutlandırması adım 6 da yapılmaktadır.
3. Eğitimi yapılacak nesne sayısı kadar dış for döngüsü başlatılıyor algoritma 5.1'de obje sayısı 3 olduğundan ilgili sayı 3 tür.
4. Eğitimi yapılacak objeye ait imge sayısı kadar iç for döngüsü başlatılıyor algoritma 5.1'de objeye ait imge 100 olduğundan bu sayı 100 olarak alınmıştır.
5. Eğitimi yapılan objeye ait imgeler döngü içerisinde  $i$  değişkenine bağlı olarak sırasıyla okutulmaktadır.
6. 5. Adımda okutulan imge 92 x 112 boyutuna dönüştürülmektedir.
7. İmge ilk olarak gri formata dönüştürülüyor daha sonra siyah beyaz, yani ikili formata dönüştürülüyor. Morfolojik yapılandırma elemanları kullanılarak net bir görüntü elde edilmeye çalışılmaktadır. Ön işlemde ve morfolojik işlemlerden geçen bir imge şekil 5.4'te görülmektedir.
8. 7. Adımdan sonra elde edilen imge ikili formatta 2. Adımda oluşturulan  $V$  matrisinin ilgili boyutuna kaydedilmektedir.



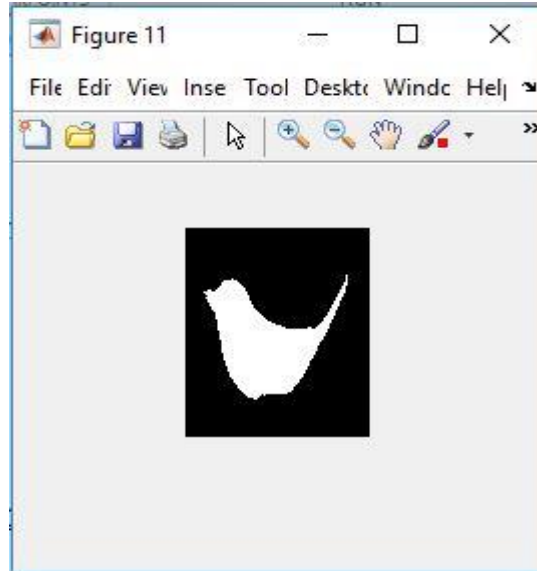
9. For döngülerinin sonu

10. Eğitimi yapılan objeler tamamlandığında  $V$  matrisini ELM'nin datasetine kaydedilmektedir.

Kuş resminin eğitim aşamasında kullanılan ilk görüntüsünün orijinal hali şekil 5.3'de görülmektedir. ELM'nin eğitimi gösteren algoritma 5.1'de bu kuş resminin 7. Adımın sonunda ön işlemden geçirilmesi sonucunda elde edilen durumu Şekil 5.4'de görülmektedir. Başka bir örnekte şekil 5.5'te görüntünün orijinal hali şekil 5.6'da ise ön işlemden sonraki hali görülmektedir.



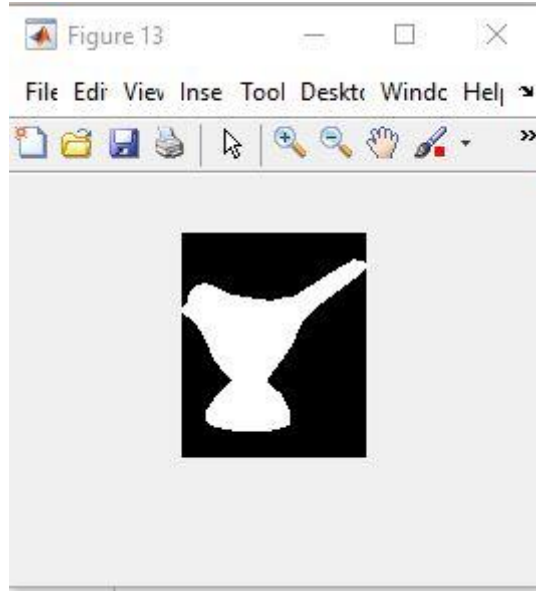
Şekil 5.3 Orijinal görüntü



Şekil 5.4 Ön işlemden sonra elde edilen görüntü



Şekil 5.5 Görüntünün orijinali



Şekil 5.6 Ön işlem sonucu elde edilen görüntü

## 5.2.ELM'nin Test Bölümü

İlk olarak segmentasyonu istenen ply formatındaki orijinal 3D görüntü programdaki 3D görüntü seç adındaki buton yardımıyla seçilir. Şekil 5.8'de segmentasyon için hazırlanan yazılımın ekran görüntüsü verilmiştir. Test süreci algoritma 5.2'de, test sürecine ait diyagram şekil 5.2'de gösterilmiştir.

### Algoritma 5.2 Test süreci

1. For i = 1 : Test Sayısı (20)
2. i. ply formatındaki 3D modelin alınması
3. i. ply formatındaki 3D modelin 2D' ye çevrimi (Algoritma 5.4)
4. i. ply formatındaki 2D resmin test verilerinin oluşturulması (Algoritma 5.3)
5. i. segmentasyonun gerçekleştirilmesi (Algoritma 5.6)
6. End i

### Algoritma 5.2'nin adımları :

1. Kullanıcının 20 adet segmentasyonu istenen görüntüyü seçmesi için döngü başlatılır.
2. Kullanıcı Şekil 5.8'deki GUI üzerindeki bir butonla sırayla segmentasyonu istenen görüntüleri seçmektedir.
3. Ply formatındaki 3D görüntü Algoritma 5.4'te anlatıldığı gibi 2D'ye çevrilmektedir.
4. 3. Adımda 2D ye dönüştürülen görüntünün ELM için test verilerini oluştur. Bu işlemler Algoritma 5.3'de gösterilmiştir.
5. 3D görüntünün segmentasyon işlemini gerçekleştir. Bu işlemler Algoritma 5.6'da anlatılmıştır.
6. Döngü sonu

### Algoritma 5.3 ELM'nin test verilerinin oluşturulması

1. NS=Algoritma 5.5 kullanılarak 2D' ye dönüştürülen imgedeki nesne sayısı bulunur
2.  $V=10304 (92 \times 112) \times NS$  ölçülerinde matris oluştur
3. Test görüntülerinin olduğu klasöre git
4. For j=1: NS
5.     a=j. resmi oku
6.     a= resim imgesini ön işlemden geçir
7.     Resim imgesini  $V(:, j)$  matrisine görüntüyü binary formatta kaydet(a)
8. End
9. V ve Etiket matrislerini ELM'nin test data bölümüne kaydet

#### Algoritma 5.3'nin adımları:

1. Algoritma 5.5'de elde edilen obje sayısını algoritma 5.3'e göre NS değişkenine atar.
2. Her bir obje için obje sayısı kadar(NS) 10304 boyutunda vektör oluşturulmaktadır. ELM'nin test verisi için kullanılacak her bir imgenin boyutu 92 x 112 olduğundan 10304 boyutunda vektör oluşturulmaktadır.
3. Algoritma 5.4'de anlatıldığı gibi, görüntülerin 2D'ye dönüştürülüp kaydedilen klasöre gidilmektedir
4. For döngüsü obje sayısına kadar devam edecek şekilde başlatılmaktadır.
5. Döngü içerisinde j değerine bağlı olarak sırayla imgeler okunmaktadır.
6. ELM'den daha iyi bir sonuç alabilmek için imgeler tekrar ön işlemden geçiriliyor. İmgelerdeki boşluklar dolduruluyor, parazitler gideriliyor, morfolojik işlemler yapılmaktadır.
7. 5.Adımdan sonra elde edilen imge 2. Adımda oluşturulan V matrisinin ilgili boyutuna kaydedilmektedir.
8. Döngü sonu
9. V matrisi ELM'nin test data bölümüne kaydedilmektedir.

#### Algoritma 5.4 3D görüntünün 2D ye dönüştürülmesi

1. I=ply dosyasını oku.(Orijinal 3D görüntü)
2. Orijinal 3D görüntüyü üç boyutlu olarak ekranda göster(I)
3. L=3D görüntüyü 2D'ye dönüştür(I)
4. 2D dönüştürülen imgeyi kaydet(L)

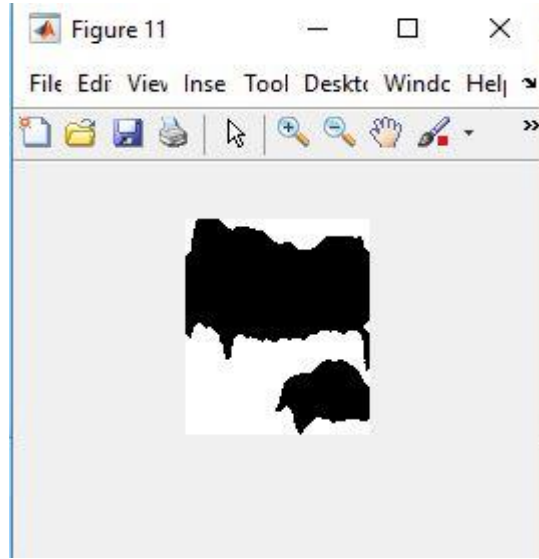
Algoritma 5.4'nin adımları:

1. Ply formatındaki üç boyutlu görüntü dosyasının okunmaktadır
2. Orijinal 3D görüntünün üç boyutlu olarak ekranda gösterilmektedir
3. Üç boyutlu görüntü iki boyuta dönüştürülüyor. İki boyuta dönüştürmek için üç boyutlu dosya 1.adımda I değişkenine alındığında I değişkeni, üç boyutlu görüntünün yükseklik(x), genişlik(y) ve derinlik(z) bilgilerini, renk bilgilerini, genişlik, derinlik, yükseklik bilgilerinin maksimum minimum değerlerini, oluşan matrisin x değerini de içeren algoritma 5.4'e göre I adında bir sınıf oluşturulmaktadır. Üç boyutlu görüntüyü iki boyuta dönüştürmek için kullanılan fonksiyona I sınıfındaki x, y ,z bilgileri ve iki boyuta dönüştüğünde imgenin boyut bilgileri (mm x nn) gönderilmektedir.

Daha sonra bu bilgilere göre derinlik bilgisi hesaplanıyor(d). İmgenin en yakın komşularını bulmada kullanılacak olan grid yapısı x, y, z bilgilerinin maksimum ve minimum değerleri kullanılarak oluşturulmaktadır. Oluşturulan grid yapının merkezi hesaplanmaktadır. En yakın komşu bulma sınıfında, grid merkezinin değeri de kullanılarak gerekli hesaplamalar yapılır, x değerinin ortalaması hesaplanır(ort). En yakın komşu değeri, derinlik bilgisi(d), imge iki boyuta dönüştüğündeki boyutları(mm x nn), x değerinin ortalaması(ort) verileri kullanılarak veri grubu oluşturulur. Daha sonra bu veri grubundaki bilgilerle yeniden yapılandırma yapılarak 2D görüntü elde edilmektedir.

4. Bu adım da elde edilen iki boyutlu görüntü test verisi olarak kullanılmak üzere kaydedilmektedir.

Şekil 5.7'de üç boyuttan iki boyuta dönüştürülüp sonra ikili görüntüye dönüştürülmüş görüntü görülmektedir.



Şekil 5.7 Orijinal üç boyutlu görüntünün iki boyuta dönüştürülmüş şekli

#### Algoritma 5.5 Nesnelerin ve sayısının bulunması

1. Görüntü= 2D ye dönüştürülen görüntüyü al
2. [label]=görüntüde bağlı bileşenleri etiketle(Görüntü)
3. konum=görüntüyü bölgelere ayır(label)
4. nesnesayisi=0
5. For i=1:boyut(konum)
6. Minimumdegerler=konum(i)
7. maksimumdegerler=konum(i)
8. nesne=görüntüyüal(i,minimumdegerler,maksimumdegerler),
9. sn=boyutlandır(objeul,[92 112])
10. elde edilen görüntüyü(sn) kaydet
11. nesnesayisi=nesnesayisi+1
12. end

#### Algoritma 5.5'in adımları:

1. Algoritma 5.4 kullanılarak iki boyuta dönüştürülerek kaydedilen test görüntü dosyası okunuyor.
2. Görüntü deki birleşik nesnelere başlangıçta etiketler. Sonra gruptaki komşu pikselleri işleyerek her bir grup bir nesneyi temsil etmek üzere yapılandırılır.
3. 2. Adımda nesnelere temsil etmek üzere gruplandırılan bölge alınmaktadır.
4. nesnesayisi adlı değişkene 0 başlangıç değeri atanıyor daha sonra obje sayısı arttıkça bu değişkenin değeri de arttırılmaktadır.
5. Görüntüdeki nesne sayısını bulmak üzere for döngüsü 3. Adımda elde edilen konum bilgisinin değerine göre başlatılmaktadır.
6. 3.adımda elde edilen konum vektörünün i. değerine göre x ve y koordinatlarının minimum değerleri bulunur.
7. 3.adımda elde edilen konum vektörünün i. değerine göre x, ve y koordinatlarının maksimum değerleri bulunur.
8. İlgili fonksiyona görüntüye ait gerekli bilgiler, koordinat verisi, maksimum, minimum değerleri verildiğinde görüntüden istenen bölge, yani nesne çıkarılmaktadır.
9. Elde edilen görüntü uygun boyutlara getirilmektedir.
10. İstenilen boyuta getirilen görüntüler sırayla kaydedilir.
11. Objeye tespit edildiğinde nesnesayisi değişkeninin değerini arttırılır.
12. Döngü sonu

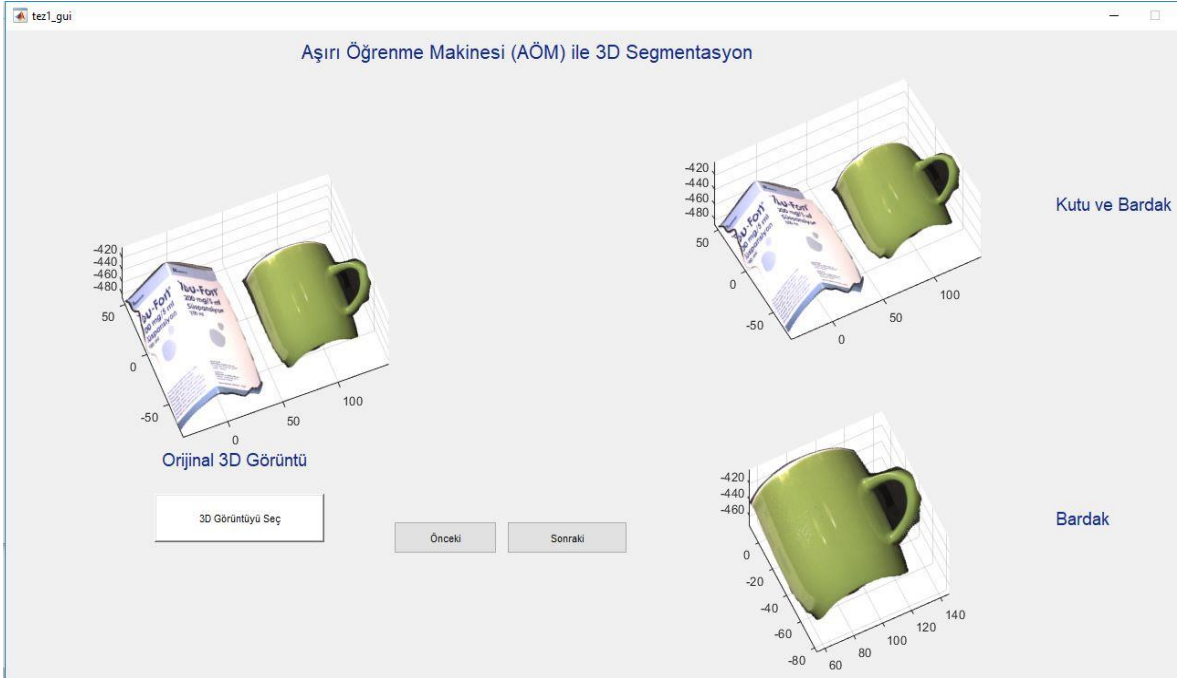
#### Algoritma 5.6 Segmentasyon işlemi

1. Sonuç=ELM() % Algoritma 5.3 kullanılarak elde edilen test verilerinin ELM'ye verilip çalıştırdıktan sonraki sonuçlar
2. Görüntü=ply dosyasını oku. %Orijinal 3D görüntü
3. Eğer Sonuç==1 %ELM nesneyi tanımış ise segmentasyonu yap
4. Konum=Görüntü içinden nesnenin konumunu bul
5. Nesne=Görüntü içinden Konum a göre nesneyi çıkart
6. Nesne adı=algoritma 5.7 kullanılarak nesnenin adını bul
7. Nesne yi ekranda göster.
8. Nesne nin adını nesnenin yanına yaz.
9. Eğer Sonuç ~1 %ELM nesneyi tanımamış ise segmentasyonu yapma
10. Nesnenin yanına "Tanınmayan nesne" ifadesini yaz

Algoritma 5.6'nın adımları:

1. Algoritma 5.3'de anlatıldığı gibi ELM'nin test verileri ve algoritma 5.1'de ELM'nin eğitiminden elde edilen öğrenme verileri kullanılarak ELM'den elde edilen sonuç verisi
2. Üç boyutlu ply formatındaki segmentasyonu istenen görüntü dosyanın okunması.
3. ELM'nin çıkarım yaptığı Sonuç değeri 1 ise segmentasyon işlemini gerçekleştir.
4. Nesnenin konumunu bulmak için ilk olarak üç boyutlu görüntü iki boyuta dönüştürülür. 3D boyuttan 2D boyuta dönüştürme işlemi Algoritma 5.4'de açıklanmıştır. İki boyuta dönüştürülen görüntü ikili formatta yani hücreleri 0 ve 1 ler den oluşmaktadır. Elde edilen görüntü For döngüsü ile satır sütunlarına bakılarak 0 lar ve 1 lerin yerleri tespit edilir. Bilindiği üzere 0 siyah yani aranan nesnenin pikselleri ne karşılık gelmektedir. 1 ise beyaza karşılık gelmekte ve buna göre nesnenin konumu tespit edilmektedir.
5. 6. Adımda konumu tespit edilen nesne 2. adımda elde edilen Görüntü içerisinde konum bilgisine göre çıkarılır.
6. Algoritma 5.7'de anlatıldığı gibi nesne adını bulunur.
7. Nesneyi ekranda göster. Şekil 5.8'de görülmektedir.
8. Nesnenin adını yan tarafına yaz. Şekil 5.8'de görülmektedir.
9. ELM'nin çıkarım yaptığı Sonuç değeri 1 değil ise segmentasyon işlemini yapılmamaktadır. Eğitimi yapılmayan bir nesnenin segmentasyonu istenmemekte ve eğitimi yapılan nesnelerin segmentasyonu istenmektedir.
10. Eğitimi yapılmamış ise nesne ekranda gösteriliyor yanına tanınmayan nesne ifadesi yazılmaktadır.





Şekil 5.8 Eğitimi yapılmış nesnelerin segmentasyonu ve isimlerinin yazılması

Algoritma 5.7 ELM'nin elde ettiği sonuçlara göre obje isimlerinin bulunması

1. [obje\_adibul\_testveri,obje\_adibul\_öğrenmeveri,obje\_adibul\_testetiket]=ELM()  
%ELM'nin çalışması sonucu elde edilen veriler
2. [Obje\_isimleri]=Obje\_isimlerini\_bul(obje\_adibul\_testveri,  
obje\_adibul\_öğrenmeveri,obje\_adibul\_testetiket)
3. Sinifadlar=load()
4. Obj\_e\_isimleri=""
5. Sinif\_degeri=sinifsaybul(obje\_adibul\_öğrenmeveri)
6. Sinifsay=sınıfbul(Sinif\_degeri)
7. Test\_say=testbul(obje\_adibul\_testetiket)
8. For i=1: Sinifsay
9. For k=1: Test\_say
10. If (obje\_adibul\_testveri(i,k)==1
11. Obj\_e\_isimleri(i)=sinifadlar(1,i)
12. end end end end
13. Obj\_eadi.text.string=Obj\_e\_isimleri(1,1)

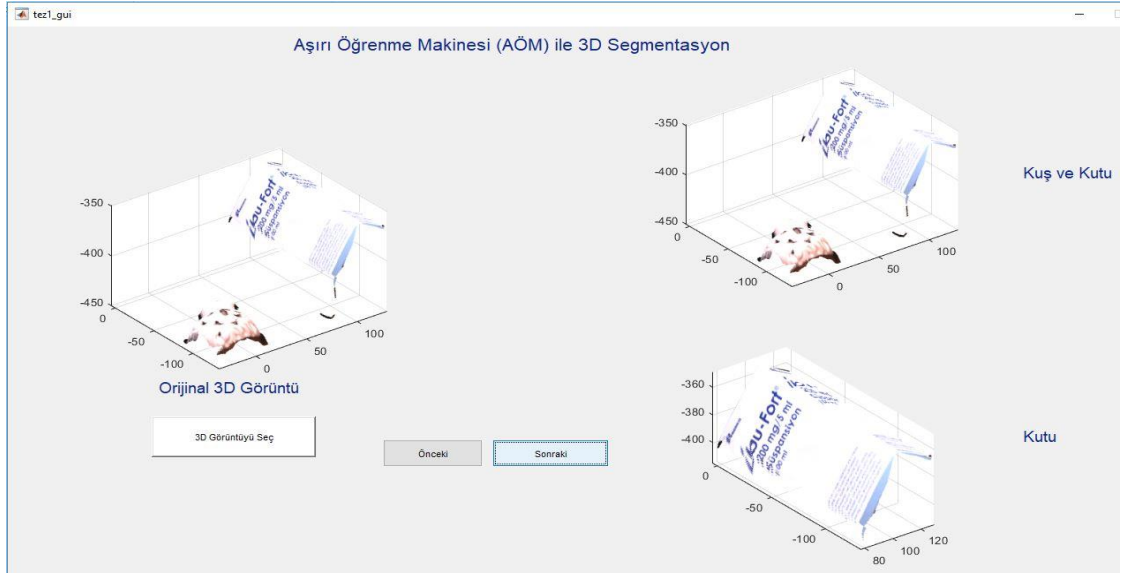
Algoritma 5.7'in adımları:

1. ELM den gelen veriler obje\_adibul\_testveri, obje\_adibul\_öğrenmeveri, obje\_adibul\_testetiket adlı değişkenlere alınmaktadır.
2. ELM'nin çalışması sonucu elde edilen sonuçlardan, eğitimi yapılan obje veya objeleri ELM tanımlayarak test-etiket adında bir matris oluşturuyor. Bu matris obje\_adibul\_testveri adıyla, öğrenme dataları obje\_adibul\_öğrenmeveri, adıyla ve test etiket verileri obje\_adibul\_testetiket adıyla Obje\_isimlerini\_bul adlı fonksiyona gönderilmektedir.
3. ELM'nin eğitimi sırasında oluşturulan obje isimleri sinifadlar adlı diziyeye yüklenmektedir.
4. Obje\_isimleri adlı bir dizi oluşturulur.
5. Öğrenme verileri kullanılarak kaç tane objenin eğitimi yapıldığını bir vektör olarak elde edilir.
6. 5. Adımda elde edilen veri kullanılarak kaç farklı sınıf olduğunu bulur.
7. Orijinal görüntüde kaç farklı obje olduğunu test etiket verilerine göre tespit eder.
8. Sinif\_say değerine göre dış for döngüsünü başlatır.
9. Test\_say değerine göre iç for döngüsünü başlatır.
10. ELM'de elde edilen test etiket verisi sonucu 1 ise eğitimi yapılan obje demektir.
11. Obje\_isimleri dizisine Objenin adını 3. Adımda oluşturulan sinifadlar adlı diziden alarak kaydediyor.
12. İf ve for döngü bloklarının sonu
13. Elde edilen isimler ilgili metin kutularını yazdırılır.

### **5.3.ELM ile Segmentasyon İçin Hazırlanan Yazılım**

ELM ile segmentasyon sürecinde ply formatındaki 3D modelin segmentasyonu gerçekleştirilerek yalnızca istenen nesne/nesnelerin 3D modelinin ayrıştırılması sağlanmaktadır. Bu süreç Algoritma 5.1'de anlatılmaktadır. Yazılım görüntüsü Şekil 5.9'da görülmektedir.

Bir sonraki objenin segmente edilen görüntüsü programdaki sonraki adlı buton yardımıyla görüntülenir ve objenin ismi görüntünün yanına yazdırılır. ELM'nin test bölümünde işlem görebilmesi için, görüntüleme teknikleri kullanılarak 2D haline getirilir. 3D görüntünün 2D ye dönüştürülme işlemi Algoritma 5.4'de anlatılmıştır. Elde edilen görüntü ELM'nin test data bölümüne iletilir. Aynı zamanda obje veya objelerin etiket verileri ELM'nin etiket data bölümüne aktarılır. ELM'nin çıkarımı sonucu orijinal 3D görüntüden obje veya objeler elde edilir.



Şekil 5.9 Önerilen yöntemle segmentasyon için hazırlanan GUI

Algoritma 5.8 3D görüntünün seçilmesi ve diğer algoritmaların çalıştırılması

1. [f, p]= dosya seçmek için pencereyi aç('\*.','Open File');
2. Dosyaad=seçilen dosyaadını oku(p,f)
3. I=üçboyutludosyayıoku(Dosyaad)
4. Orijinal 3D görüntüyü üç boyutlu olarak ekranda göster(I)
5. Pt=3D görüntüyü 2D'ye dönüştür.
6. Test veri algoritması()
7. [konum, renk]=segmentasyon(I, Pt, 1)
8. Segmente\_obje(konum,renk) segmente edilen objeyi verilerine göre görüntüle
9. [isimbul\_testlabel\_elm,isimbul\_trainlabel,isimbul\_testlabel]=ELM()
10. [Obj\_e\_isimleri]= Obj\_e\_isimlerini\_bul  
(isimbul\_testlabel\_elm,isimbul\_trainlabel,isimbul\_testlabel)
11. Text.string=Obj\_e\_isimleri(1,1)

Algoritma 5.8'nin adımları:

1. Kullanıcı segmentasyonunu istediği görüntüyü açılan pencereden seçer.
2. Seçilen dosyanın adı ve yol bilgileri kullanılarak dosya adı belirlenir.

3. Belirlenen dosya okunarak I deęişkenine atanır.
4. Seçilen üç boyutlu görüntü ekranda gösterilir.
5. Üç boyutlu görüntü iki boyuta dönüştürülür. 3D görüntünün 2D'ye dönüştürülmesi işlemi algoritma 5.4'de gösterilmiştir.
6. Test veri algoritmasını çalıştır. Test veri algoritması algoritma 5.3'de anlatılmıştır.
7. Segmentasyon algoritmasını çalıştır. Algoritma sonucu dönen deęerler konum ve renk adındaki vektörlerde tutulmaktadır. Segmentasyon algoritması algoritma 5.6'da anlatılmıştır.
8. 7. Adımda segmentasyon algoritması sonucu elde edilen verilerle segmente edilmiş objeyi ekranda gösterilir.
9. ELM algoritması çalıştırıldıktan sonra dönen deęerler isimbul\_testlabel\_elm, isimbul\_trainlabel, isimbul\_testlabel adlı vektörlerde tutulmaktadır. Bu işlem algoritma 5.7'de anlatılmıştır.
10. 9. Adımda elde edilen veriler Obje\_isimlerini\_bul adlı fonksiyona gönderilir. Daha sonra fonksiyondan dönen veriler Obje\_isimleri adlı vektörde tutulur. Bu vektörde segmentasyon sonucu elde edilen objelerin isimleri bulunmaktadır. Söz konusu işlem Algoritma 5.7'de anlatılmıştır.
11. Bir önceki adımda elde edilen obje isimleri, objelerin görüntülerinin yanındaki metin kutularına yazdırılarak ekranda gösterilir.

Algoritma 5.9 ilk objeden sonra dięer objelerin görüntülenmesi

- |  |
|--|
| <ol style="list-style-type: none"><li>1. nesnesay=algoritma 5.5'den elde edilen nesne sayısı</li><li>2. [konum, renk]=segmentasyon(I, Pt, nesnesay)</li><li>3. Segmente_obje(konum,renk) segmente edilen objeyi verilerine göre görüntüle</li><li>4. Text.string=Obje_isimleri(1,Objesay)</li><li>5. nesnesay= nesnesay+1</li><li>6. Objesay=objesay+1</li></ol> |
|--|

Algoritma 5.9'un adımları:

1. Nesnelerin sayısının bulunması işlemi Algoritma 5.5'de anlatılmıştır.
2. Segmentasyon adlı algoritma çalıştırılır. I ve Pt deęerleri Algoritma 5.3'de elde edilmişti aynı deęerler burada da kullanılmaktadır. Segmentasyon algoritması Algoritma 5.6'da anlatılmıştır.

3. Bir önceki adımda elde edilen konum ve renk vektörlerindeki veriler kullanılarak obje ekranda görüntüleniyor.
4. Objenin adı obje\_isimleri adlı vektörden objesay değişkenine göre alınarak ilgili metin kutusuna yazdırılıyor.

#### 5.4. Kullanılan Donanımlar

3D görüntü elde etmek için FUEL3D adlı cihaz kullanılmıştır. Cihaza ait görüntü Şekil 5.10'da gösterilmiştir. Fotoğraftaki görünen ışıklar led kılavuz ışıklarıdır. Her birinin yanında bir Xenon flaş, merkezdeki beyaz çerçevede bir RGB kamera var. Bir diğer kamera alt orta kısımda hafif bir açı ile altında yer almaktadır. Her iki kamerada 3.5 megapiksel özelliğe sahiptir.



Şekil 5.10 3D görüntü almak için kullanılan cihaz



Şekil 5.11 Deneysel çalışmalardan görüntü

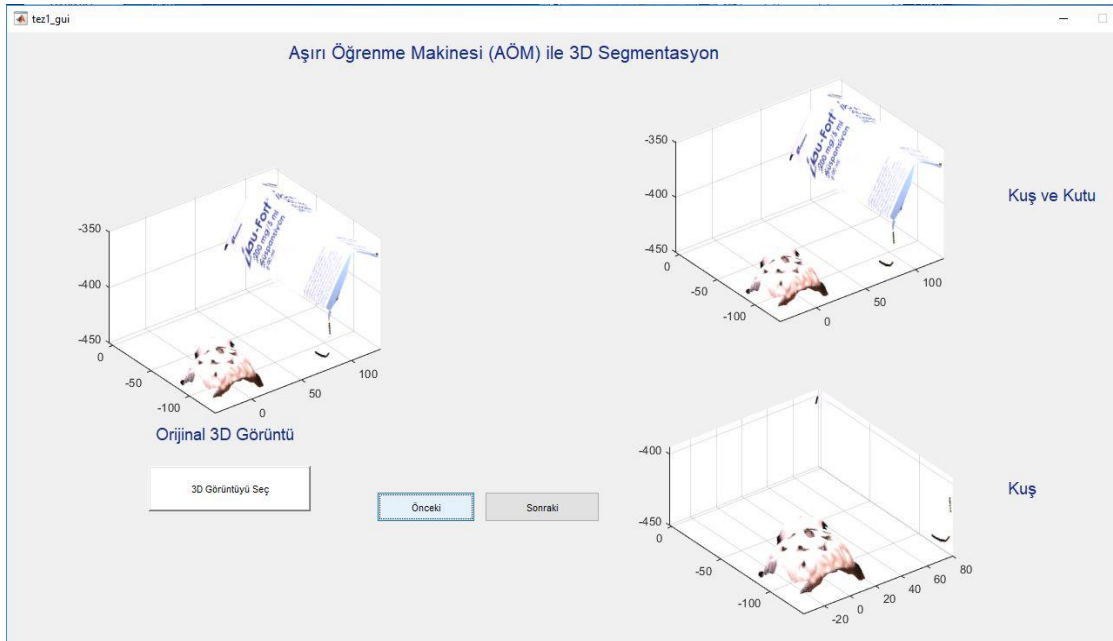
Bazı 3D görüntü alan sistemlerde projeksiyon cihazı ve diğer ekipmanlar kullanıldığından sistemi kullanmak zaman alıcı ve yavaş olabilmektedir. Bu cihaz ise fotoğraf makinesi kullanır gibi oldukça kolay ve hızlı bir şekilde ve kullandığı özel bir yazılım sayesinde 3D görüntü oluşturmaktadır. Cihaz usb kablosu ile bilgisayara bağlandıktan sonra Fuel3D Studio adlı program ile 3D görüntü yakalama işlemi yapabilmektedir. Deneysel çalışmalar sırasında cihazla 3D görüntü alma işlemi şekil 5.11’de gösterilmiştir.

### 5.5. Deneysel Çalışmalar

Bu çalışmada eğitimi yapılan üç adet obje, biblo kuş, ilaç kutusu ve kupa bardak ELM ye öğretilmiştir. İlk olarak biblo kuş objesine ait benzer 100 adet biblo kuş fotoğrafı görüntü işleme tekniklerine tabi tutularak gerekli filtrelemelerle ön işlem yapıldı. Sonra Şekil 5.1’de gösterildiği gibi eğitim süreci gerçekleştirilmiştir. Yine aynı şekilde kupa bardağın eğitimi de 100 adet kupa bardak görüntüsü, ilaç kutusunun eğitiminde de 100 adet ilaç kutusu görüntüsü kullanılmıştır. Eğitim sonuçları ELM’nin datasetine eklendi.

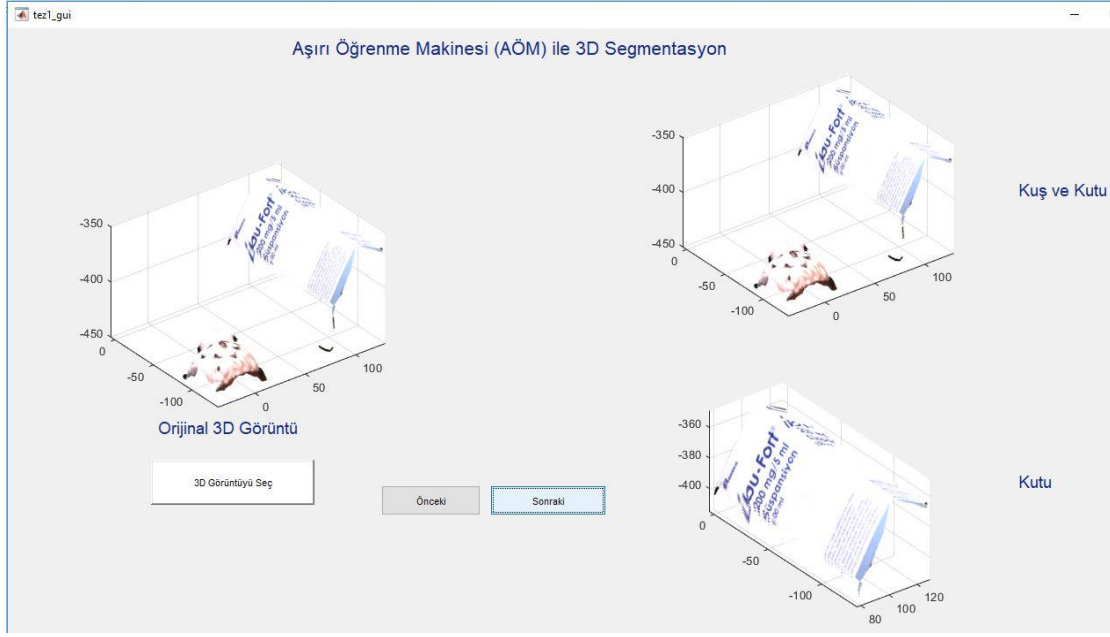
Şekil 5.12 ve Şekil 5.13’de görüldüğü gibi, önerilen segmentasyon yöntemi ile, eğitilen objelerin her ikisinin de olduğu 3D bir görüntüden, her iki objede orijinal durumlarına en yakın şekilde 3D olarak başarılı bir şekilde segmente edilmektedir.

Her iki objenin de eğitimi yapıldığından segmentasyon sonucunda objelerin adları da görüntünün yanında görülmektedir. Önceki ve sonraki butonlarıyla objelerin 3D segmentasyon sonucu ekranda gösterildiğinde nesnelerin isimleri de ekranda görülmektedir. Şekil 5.12’de kuş görüntüsünün yanında kuş adı görülmektedir. Şekil 5.13’de kutu görüntüsünün yanında kutu adı görülmektedir.

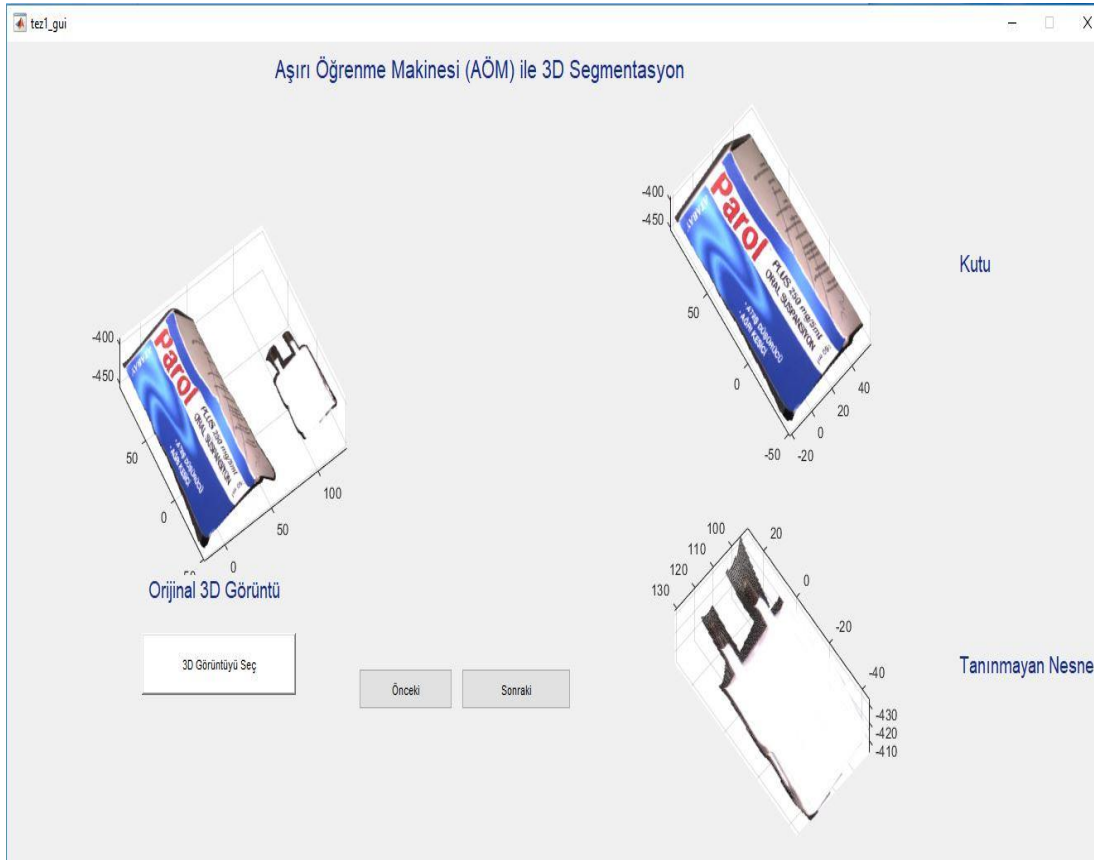


Şekil 5.12 Eğitim yapılan iki objenin segmentasyonu

Eğitimi yapılan objelerden kutunun bulunduğu, eğitimi yapılmayan yani ELM’ye öğretilmeyen başka bir objenin bulunduğu bir 3D görüntü önerilen yöntem ile segmentasyon işlemi yapıldığında, kutuyu bularak 3D görüntüsünün yan tarafına kutu adını yazdığı, eğitimi yapılmayan nesneye ise tanınmayan nesne olarak adını görüntünün yan tarafına yazdığı şekil 5.13’de görülmektedir.



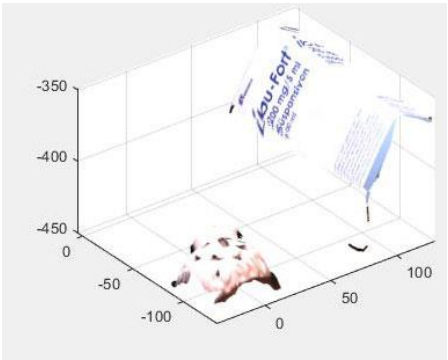
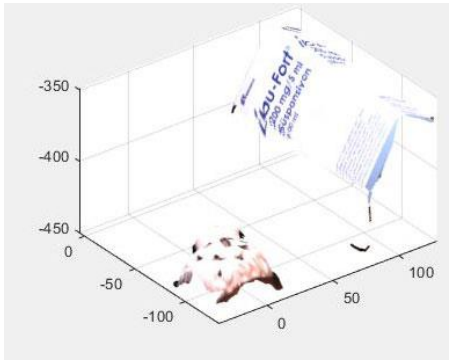
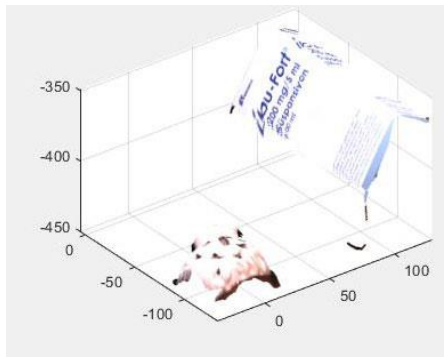
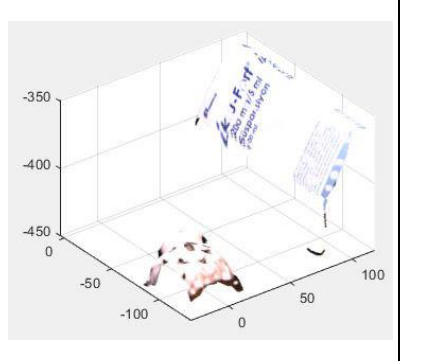
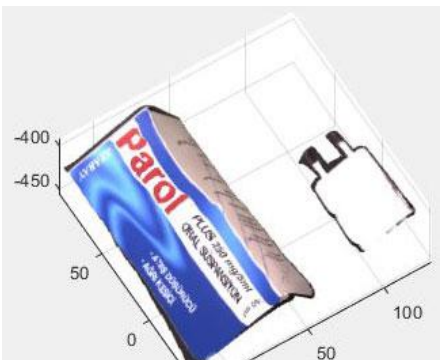

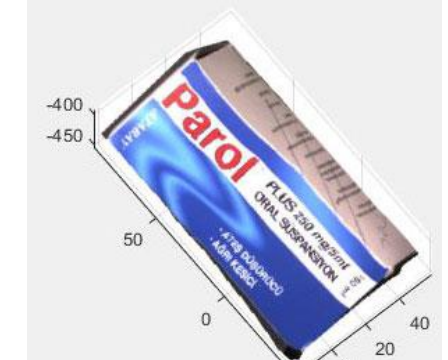
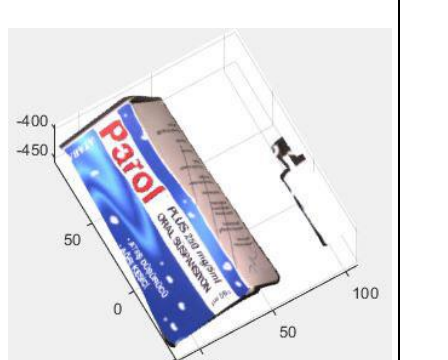
Şekil 5.13 Eğitim yapılan iki objenin segmentasyonu



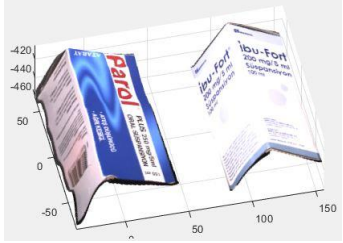
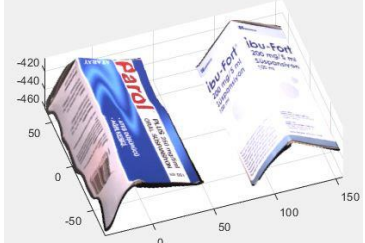
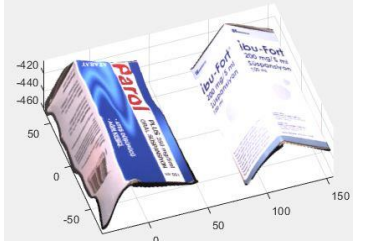
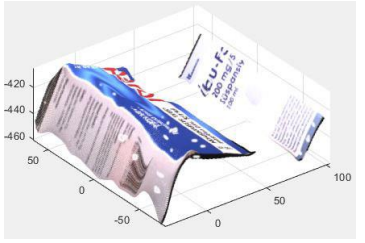
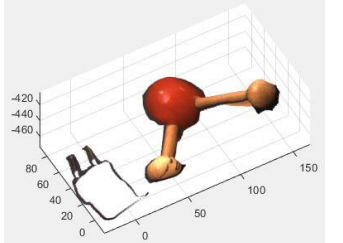
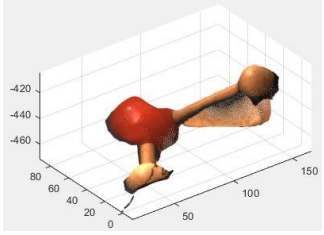
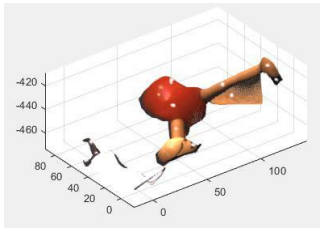
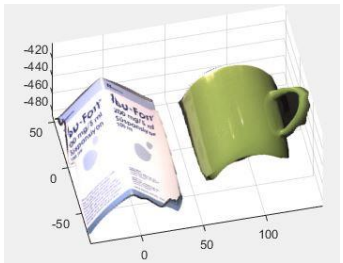
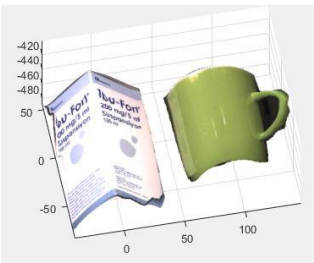
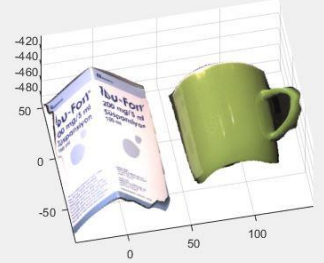
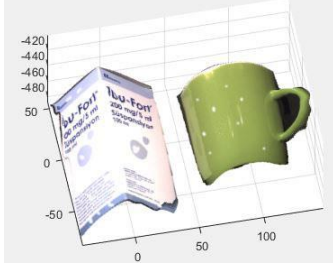
Şekil 5.14 Eğitimi yapılan obje ile eğitimi yapılmayan obje segmentasyon sonucu



Tablo 5.1 a) ELM, YSA, FCM segmentasyon sonuçları

	Orijinal 3D	ELM	YSA	FCM
1				
2				

Tablo 5.1 b) ELM, YSA, FCM segmentasyon sonuçları

	Orjinal 3D	ELM	YSA	FCM
3				
4		Eğitimi yapılmayan iki nesne olduğu için beklendiği gibi segmentasyon gerçekleşmedi		
5				

1. test sonucunda ELM ve YSA'ya kuş ve ilaç kutusu nesneleri eğitimi verildiğinden bu nesneleri başarılı bir şekilde segmente etmişlerdir. FCM'nin segmentasyon işlemi sonucunda ise 3D modelin üzerinde boşluklar olduğu görülmektedir. Bu durum FCM'nin 2D segmentasyonda görülen başarısızlığın 3D modelleme üzerindeki etkisini göstermiştir.

2. test sonucunda ELM segmentasyon işlemi sonucunda beklendiği gibi ilaç kutusunu çıkartmaktadır çünkü diğer objenin yani şarj cihazının eğitimi yapılmadığı için bu objenin segmentasyon işlemi sonucunda çıkmaması gerekmektedir. YSA da aynı sonucu buldu. FCM ise iki objenin de segmentasyonunu yaparak başarısız oldu.

3. test sonucunda ELM ve YSA istenen segmentasyon sonucunu elde ederken FCM segmentasyon sonucunu bozuk bir şekilde çıkarmıştır. FCM'nin 2D segmentasyondaki başarısızlığının sonucunda 3D modellemişinin de başarısızlığına neden olmuştur.

4. test işleminde iki objenin de eğitimi yapılmadığı için beklendiği gibi ELM segmentasyon işlemi sonucunda herhangi bir sonuç çıkarmamıştır. YSA ise objelerin segmentasyon sonucunu çıkarmaması gerekirken sonuç çıkarmıştır. FCM de segmentasyon işlemini hatalı olarak çıkarmıştır.

5. test iki objesinde eğitimi yapıldığı için ELM segmentasyon işlemi sonucunda iki objenin görüntüsünü çıkarmış, YSA da segmentasyon işlemini gerçekleştirmiştir. FCM hatalı segmentasyon işlemi yapmıştır.

Yukarıda ayrıntısı sunulan 5 adet test işlemine ek olarak 15 adet test işlemi de yapılmıştır. 3D modelleme kalitesini ortaya çıkarmak için her bir test sonucunda elde edilen 3D modelleme sonucu 2D'ye çevrildikten sonra Jaccard İndeksi (JI) ile segmentasyon başarımlarını performansları ortaya konmuştur. JI testiyle, olması gereken segmentasyon sonucuyla uygulanan yöntemin segmentasyon sonucu karşılaştırılmakta ve 0-1 arasında bir değer döndürülmektedir. 1 değeri en ideal segmentasyon sonucunu, 0 ise en başarısız segmentasyon sonucunu göstermektedir[28]. Tablo 5.2'de 20 adet test sonucuna ait JI değerlerine ait tablo sunulmuştur. Tablo 5.3'te ise bu tablo üzerinde gerçekleştirilmiş istatistiksel analizler sunulmuştur. İlgili tablodan görülebildiği gibi en ideal sonuçlar önerilen yaklaşımla elde edilmiştir.

Tablo 5.2 Üç yaklaşıma ait Jaccard Index sonuçları

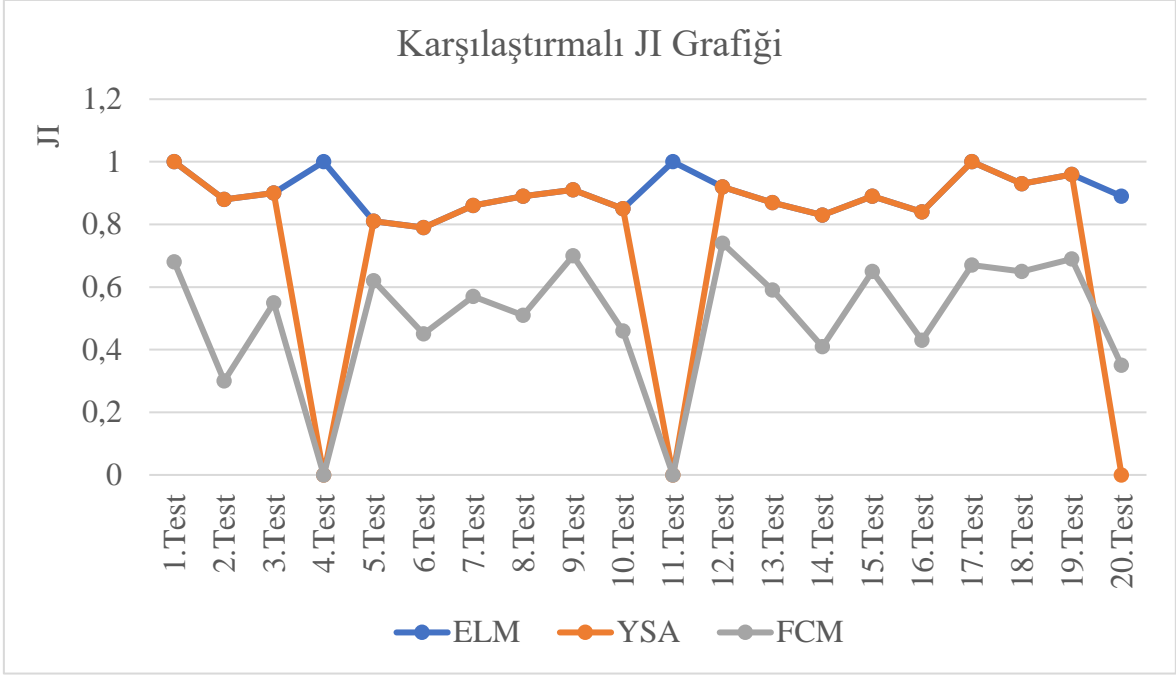
	<b>ELM</b>	<b>YSA</b>	<b>FCM</b>
1.Test	1	1	0,68
2.Test	0,88	0,88	0,30
3.Test	0,90	0,90	0,55
4.Test	1	0	0
5.Test	0,81	0,81	0,62
6.Test	0,79	0,79	0,45
7.Test	0,86	0,86	0,57
8.Test	0,89	0,89	0,51
9.Test	0,91	0,91	0,70
10.Test	0,85	0,85	0,46
11.Test	1	0	0
12.Test	0,92	0,92	0,74
13.Test	0,87	0,87	0,59
14.Test	0,83	0,83	0,41
15.Test	0,89	0,89	0,65
16.Test	0,84	0,84	0,43
17.Test	1	1	0,67
18.Test	0,93	0,93	0,65
19.Test	0,96	0,96	0,69
20.Test	0,89	0	0,35

Tablo 5.3 JI sonuçlarına ait istatistiksel sonuçlar

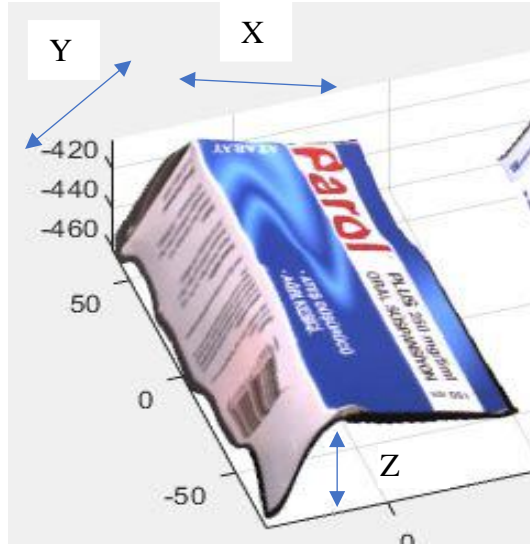
	<b>ELM</b>	<b>YSA</b>	<b>FCM</b>
<b>Min</b>	0,79	0	0
<b>Max</b>	1	1	0,74
<b>Aritmetik Ortalama</b>	0,901	0,756	0,501

Üç yöntemle gerçekleştirilen segmentasyon sonuçlarının kalitesini ortaya koymak için gerçekleştirilen JI analizine ek olarak boyut analiz işlemi de gerçekleştirilmiştir. Bu test işleminde her bir yöntemle elde edilen 3D modelin X, Y, Z olarak gösterilen kenar uzunlukları ile gerçek kenar uzunlukları arasındaki mutlak hata tespit edilmektedir. Bu ölçülendirmeler Şekil 5.15’de görülmektedir. Mutlak hata denklem 5.1’e göre hesaplanmaktadır.

$$| X | = | X_{\text{Gerçek}} - X_{\text{Ölçülen}} | \quad (5.1)$$



Şekil 5.15 Karşılaştırmalı JI Grafiği



Şekil 5.16 3D model üzerinde ölçülendirmeler

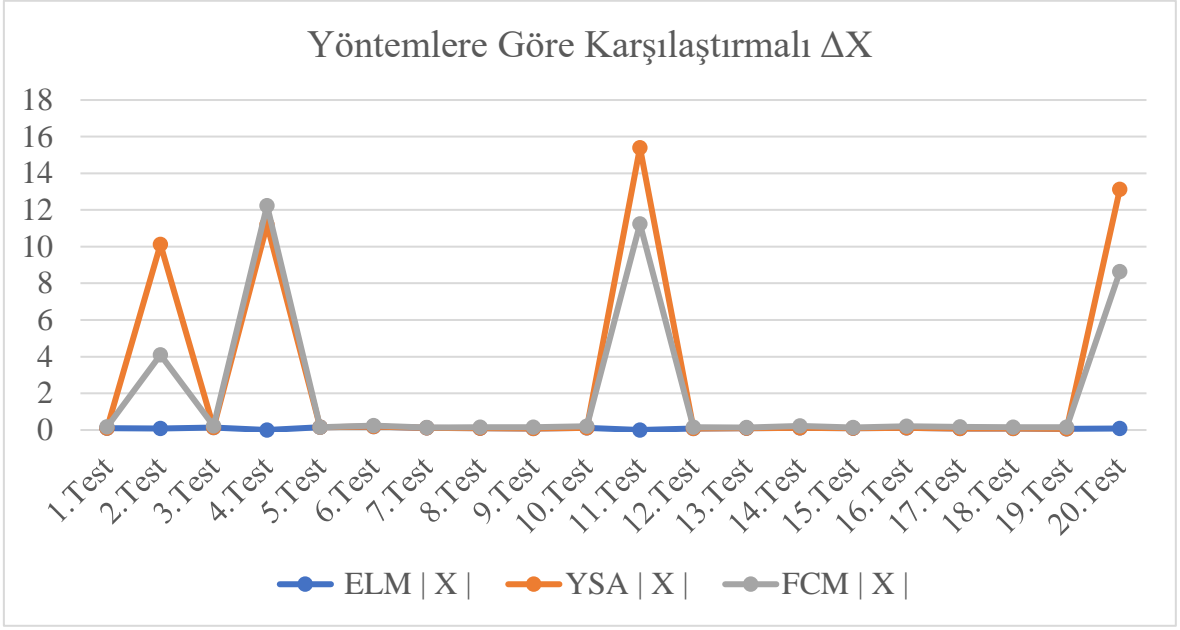
Test sürecinde elde edilen 20 adet 3D model üzerinde mutlak hata analizi gerçekleştirilmiş olup elde edilen sonuçlar Tablo 5.4’ de verilmiştir. Bu hata sonuçları üzerinde gerçekleştirilen istatistiksel analizler ise Tablo 5.5’ de sunulmuştur. Tablo 5.5’ den görülebildiği gibi en düşük mutlak hata sonuçları önerilen ELM yaklaşımı ile elde edilmiştir. Buda önerilen ELM tabanlı segmentasyon yönteminin çok başarılı bir şekilde 3D model segmentasyonunu gerçekleştirdiği ispatlanmıştır.

Tablo 5.4 Hata farkları

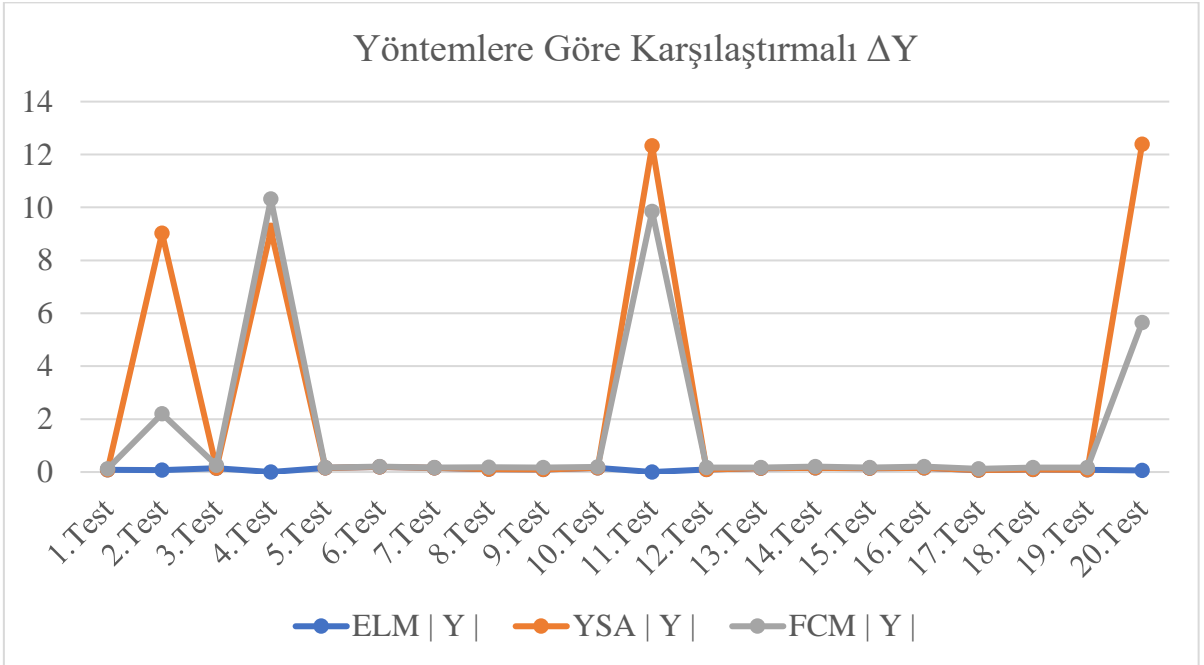
	ELM			YSA			FCM		
	X	Y	Z	X	Y	Z	X	Y	Z
1.Test	0,091	0,078	0,053	0,091	0,078	0,053	0,168	0,124	0,095
2.Test	0,089	0,077	0,046	0,089	0,077	0,046	4,102	2,204	3,201
3.Test	0,138	0,142	0,089	0,138	0,142	0,089	0,213	0,257	0,121
4.Test	0	0	0	11,21	9,130	10,023	12,243	10,314	11,143
5.Test	0,148	0,162	0,117	0,148	0,162	0,117	0,153	0,172	0,126
6.Test	0,185	0,192	0,168	0,185	0,192	0,168	0,235	0,203	0,159
7.Test	0,110	0,152	0,098	0,110	0,152	0,098	0,140	0,168	0,120
8.Test	0,090	0,110	0,085	0,090	0,110	0,085	0,155	0,175	0,130
9.Test	0,085	0,101	0,080	0,085	0,101	0,080	0,149	0,169	0,121
10.Test	0,112	0,156	0,101	0,112	0,156	0,101	0,211	0,198	0,181
11.Test	0	0	0	15,403	12,325	10,85	11,235	9,854	8,698
12.Test	0,081	0,099	0,078	0,081	0,099	0,078	0,147	0,167	0,119
13.Test	0,105	0,149	0,095	0,105	0,149	0,095	0,139	0,166	0,118
14.Test	0,115	0,159	0,108	0,115	0,159	0,108	0,215	0,203	0,197
15.Test	0,101	0,144	0,092	0,101	0,144	0,092	0,136	0,163	0,117
16.Test	0,109	0,154	0,101	0,109	0,154	0,101	0,211	0,201	0,195
17.Test	0,081	0,074	0,049	0,081	0,074	0,049	0,171	0,126	0,098
18.Test	0,079	0,094	0,071	0,079	0,094	0,071	0,151	0,172	0,123
19.Test	0,062	0,089	0,067	0,062	0,089	0,062	0,149	0,163	0,119
20.Test	0,076	0,063	0,032	13,132	12,389	10,368	8,632	5,654	4,863

Tablo 5.5 Hata farklarına ait istatistiksel sonuçlar

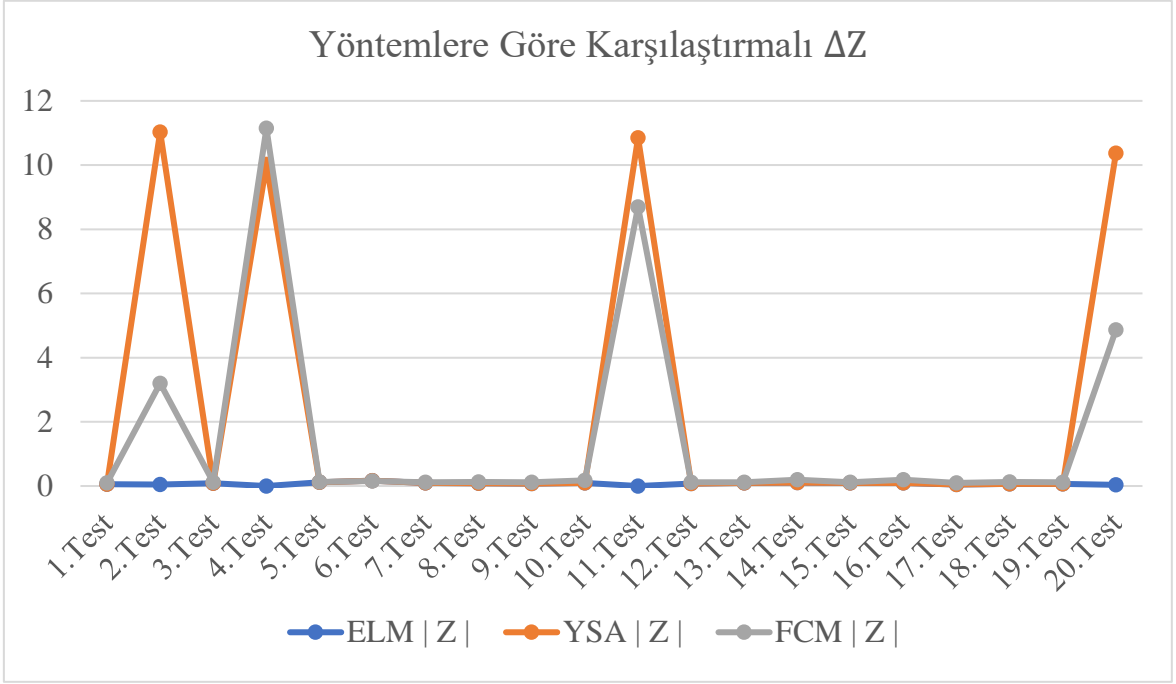
	ELM			YSA			FCM		
	X	Y	Z	X	Y	Z	X	Y	Z
<b>Min</b>	0	0	0	0,062	0,074	0,049	0,136	0,124	0,095
<b>Mak</b>	0,185	0,192	0,168	15,403	12,389	11,032	12,243	10,314	11,143
<b>Aritmetik Ortalama</b>	0,092	0,109	0,076	2,578	2,246	2,186	1,947	1,542	1,502



Şekil 5.17 Yöntemlere göre karşılaştırmalı  $\Delta X$  grafiği



Şekil 5.18 Yöntemlere göre karşılaştırmalı  $\Delta Y$  grafiği



Şekil 5.19 Yöntemlere göre karşılaştırmalı  $\Delta Z$  grafiği



## 6. SONUÇ

Görüntü işleminin önemli adımlarından biri segmentasyon işlemidir. 3D görüntülerinin derinlik bilgisi de içermesi görüntü analizi yapan sayısal sistemler için önemlidir. Nesnelerin eğitimi için tek gizli katmana sahip ileri beslemeli giriş ağırlıkları rastgele çıkış ağırlıkları ise analitik olarak hesaplanan bir YSA modeli olan ELM algoritması kullanılmıştır. Önerilen yöntemle 3D segmentasyon işleminin başarılı bir şekilde yapıldığı deneysel sonuçlar üzerinde yapılan istatistiksel analizlerle ispatlanmıştır. Önerilen yöntemin üstünlüğünü göstermek için YSA ve FCM ile karşılaştırılmış ve bu iki yönetime göre çok daha üstün bir şekilde 3D segmentasyon gerçekleştirdiği ispatlanmıştır. Önerilen yöntemin tıp, eğitim, güvenlik gibi alanlarda ihtiyaç duyulan 3D segmentasyon işlemlerinde yararlı olacağı düşünülmektedir.

## KAYNAKLAR

- [1] Öztemel, E., *Yapay Sinir Ağları*. İstanbul: Papatya Yayıncılık Eğitim (2016).
- [2] Huang, G.B, Zhu, Q. Y. ve Siew, C. K. “Extreme Learning Machine: Theory and Applications”, *Neurocomputing*, C. 70, Sayı 1–3, Ss. 489–501 (2006).
- [3] Kaya, Y., ve Tekin,R. “Epileptik Nöbetlerin Tespiti İçin Aşırı Öğrenme Makinesi Tabanlı Uzman Bir Sistem”, *BilişiteknolojiLeridergiSi*, C. 5, Sayı 2, Ss. 33–39 (2012).
- [4] Ertuğrul, O. F. ve Recognition, A. “Aşırı Öğrenme Makineleri İle Biyolojik Sinyallerin Gizli Kaynaklarına Ayırıştırılması ( Blind Signal Separation in Biological By Extreme Learning Machines ) Aşırı Öğrenme Makineleri İle Biyolojik Sinyallerin Gizli Kaynaklarına Ayırıştırılması”, Sayı June 2016 (2018).
- [5] Zhang,K., Lu,W. ve Marziliano, P. “The Unified Extreme Learning Machines and Discriminative Random Fields for Automatic Knee Cartilage and Meniscus Segmentation From Multi-Contrast Mr Images”, *Mach. Vis. Appl.*, C. 24, Sayı 7, Ss. 1459–1472 (2013).
- [6] An, L. ve Bhanu, B. “Image Super-Resolution By Extreme Learning Machine”, *Icip*, C. 1, Sayı 1, Ss. 2209–2212 (2012).
- [7] Lu,B., Wang, Yuan,G. Y., ve Han, D. “Neurocomputing Semantic Concept Detection for Video Based On Extreme Learning Machine”, C. 102, Ss. 176–183 (2013).
- [8] Achary, T., ve Ajoy,K. R., *Image Processing Principles and Applications*. New Jersey: John Wiley & Sons, Inc (2005).
- [9] “Renkli Görüntü İşleme”, 2018. [Çevrimiçi]. Available At: [Http://Homepages.İnf.Ed.Ac.Uk/Rbf/Cvonline/Local\\_Copies/Owens/Lect14/Lecture12.Html](http://Homepages.İnf.Ed.Ac.Uk/Rbf/Cvonline/Local_Copies/Owens/Lect14/Lecture12.Html). [Erişim: 15.08.2018].
- [10] “Renk Bitleri”, 2018. [Çevrimiçi]. Available At: [Https://Edoras.Sdsu.Edu/Doc/Matlab/Toolbox/İmages/Color3.Html](https://Edoras.Sdsu.Edu/Doc/Matlab/Toolbox/İmages/Color3.Html). [Erişim: 15.08.2018].
- [11] “Mathworks-Graytresh”, *Mathworks*, 2018. [Çevrimiçi]. Available At: [Https://Www.Mathworks.Com/Help/İmages/Ref/Graythresh.Html](https://Www.Mathworks.Com/Help/İmages/Ref/Graythresh.Html). [Erişim: 12.08.2018].
- [12] “Mathworks-Otsu”, *Mathworks*, 2018. [Çevrimiçi]. Available At: [Https://Www.Mathworks.Com/Help/İmages/Ref/Otsuthresh.Html?S\\_Tid=Doc\\_Ta](https://Www.Mathworks.Com/Help/İmages/Ref/Otsuthresh.Html?S_Tid=Doc_Ta). [Erişim: 10.08.2018].
- [13] “Mathworks-İmwrite”, *Mathworks*, 2018. [Çevrimiçi]. Available At: [Https://Www.Mathworks.Com/Help/Matlab/Ref/İmwrite.Html](https://Www.Mathworks.Com/Help/Matlab/Ref/İmwrite.Html). [Erişim: 12.08.2018].
- [14] “Mathworks - Ply Biçimi”, *Mathworks*, 2018. [Çevrimiçi]. Available At: [Https://www.Mathworks.Com/Help/Vision/Ug/The-Ply-Form](https://www.Mathworks.Com/Help/Vision/Ug/The-Ply-Form) [Erişim:24.09.2018]

- [15] “Ply Dosya Biçimi”, 2018. [Çevrimiçi]. Available At: [https://www.cc.gatech.edu/projects/large\\_models/ply.html](https://www.cc.gatech.edu/projects/large_models/ply.html). [Erişim: 24.09.2018].
- [16] Zhou, H., Wu, J., ve Zhang, J., *Digital Image Processing: Part I*. Huiyu Zhou, Jiahua Wu & Jianguo Zhang (2010).
- [17] Gonzalez R. C., ve Woods, R. E. *Digital Image Processing*. Pearson Prentice Hall (2008).
- [18] Haykin, S., *Neural Networks and Learning Machines*. New Jersey: Pearson Education (2009).
- [19] Canan, S., “İnsan Sinir Sistemi-3”, 2018. [Çevrimiçi]. Available At: <https://www.sinancanan.net/insan-sinir-sistemi-3/>. [Erişim: 24.06.2018].
- [20] Sons,W. J.,inc. 2000.
- [21] Hagan, M. T., *Neural Network Design*. Oklahoma (2014).
- [22] Bezdek, J. C., Ehrlich, R., ve Full, W., “Fcm: The Fuzzy C-Means Clustering Algorithm”, *Comput. Geosci.*, C. 10, Sayı 2–3, Ss. 191–203 (1984).
- [23] Performance,A., Of, D., ve Algorithms, F. C., “K-Means, K-Medoids Ve Bulanik C-Means Algorİtmalarının Uygulamalı Olarak Performanslarının TespİTi”, *İstanbul Ticaret Üniversitesi Fen Bilim. Derg.*, Ss. 31–45 (2007).
- [24] Mathur, N., Glesk,I., ve Buis, A., “Comparison Of Adaptive Neuro-Fuzzy İference System (Anfis) And Gaussian Processes For Machine Learning (Gpml) Algorithms for The Prediction of Skin Temperature İn Lower Limb Prostheses”, *Med. Eng. Phys.*, C. 38, Sayı 10, Ss. 1083–1089 (2016).
- [25] Suparta, W., ve Alhasa, K. M., “Modeling Of Tropospheric Delays Using Anfis”, Sayı 2009, Ss. 5–19 (2016).
- [26] Tang, F., Adam, L., ve Si, B., “Group Feature Selection With Multiclass Support Vector Machine”, *Neurocomputing* (2018).
- [27] Ayhan,S., “Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi”, C. 9, Sayı 1, Ss. 175–198 (2014).
- [28] Sert, E., ve Avcı, D., “Brain Tumor Segmentation Using Neutrosophic Expert Maximum Fuzzy-Sure Entropy and Other Approaches”, *Biomed. Signal Process. Control*, C. 47, Ss. 276–287 (2019).

## **ÖZGEÇMİŞ**

### **Kişisel Bilgiler**

Adı, soyadı : Ertuğrul KAYA  
Uyruğu : T.C.  
Doğum tarihi ve yeri : 26.01.1977 Kahramanmaraş  
Medeni hali : Evli  
Telefon :  
e-posta :kayaertugrul@ttmail.com

### **Eğitim**

<b>Derece</b>	<b>Eğitim Birimi</b>	<b>Mezuniyet tarihi</b>
Lisans	Sakarya Üniversitesi Teknik Eğitim Fak. Elektronik Bilgisayar Bölümü	1999
Lise	K.Maraş Batı Park Anadolu Teknik Lisesi	1994

### **İş Deneyimi**

<b>Yıl</b>	<b>Yer</b>	<b>Görev</b>
2000-2001	Kozan Teknik Lisesi	Kozan/Adana Elektronik Bilg. Öğrt.
2001-2004	Alanya Mesleki Eğm. Merk	Alanya/Antalya Elektronik Bilg. Öğrt.
2004-2009	Antalya And. Tek. Lisesi	Merkez/Antalya Elektronik Bilg. Öğrt.
2009-	Ş. İdari. Ataşe. And. Tekn. L.	Merkez/K.Maraş Bilgisayar Öğrt.

### **Yabancı Dil**

İngilizce

### **Yayınlar**

1. Kaya E., Sert E. Extreme Learning Machine-ELM Algoritması ile 3D Segmentasyon İşlemi, Sözlü Bildiri, 6th World Conference on Innovation & Computer Science Kongresi,12-14 Mayıs 2016, Kemer/Antalya

### **Hobiler**

Kayak, Yüzme, Uzay araştırmaları, Astronomi